Differentially private exact recovery for stochastic block models

Dung Nguyen [1] **Anil Vullikanti** [1]

Abstract

Stochastic block models (SBMs) are a very commonly studied network model for community detection algorithms. In the standard form of an SBM, the n vertices (or nodes) of a graph are generally divided into multiple pre-determined communities (or clusters). Connections between pairs of vertices are generated randomly and independently with pre-defined probabilities, which depend on the communities containing the two nodes. A fundamental problem in SBMs is the recovery of the community structure, and sharp information-theoretic bounds are known for recoverability for many versions of SBMs.

Our focus here is the recoverability problem in SBMs when the network is private. Under the edge differential privacy model, we derive conditions for exact recoverability in three different versions of SBMs, namely Asymmetric SBM (when communities have non-uniform sizes), General Structure SBM (with outliers), and Censored SBM (with edge features). Our private algorithms have polynomial running time w.r.t. the input graph's size, and match the recovery thresholds of the non-private setting when $\epsilon \to \infty$. In contrast, the previous best results for recoverability in SBMs only hold for the symmetric case (equal size communities), and run in quasi-polynomial time, or in polynomial time with recovery thresholds being tight up to some constants from the non-private settings.

1. Introduction

A very common first step in the analysis of networked data in numerous applications, and unsupervised machine learning is community detection or clustering, i.e., partitioning the network into "well-connected" communities, e.g., (Blondel et al., 2008; Girvan & Newman, 2002; Holland et al., 1983) (see survey by (Fortunato, 2010)). There is limited theoretical understanding of community detection, since these notions are very problem-specific. One exception is the stochastic block model (SBM) (Holland et al., 1983), a probabilistic generative model with well-defined communities, making them very amenable from a theoretical perspective. Therefore, SBMs have been extensively studied in network science, and have become standard test benches for community detection algorithms.

In the standard form of an SBM, the n vertices (or nodes) of a graph are generally divided into multiple pre-determined communities (or clusters). Connections between pairs of vertices are generated randomly and independently with pre-defined probabilities, which depend on the communities containing the two nodes; the Erdős-Rényi model is a special case of SBM with a single cluster. The simplest type of SBM is a Binary Symmetric SBM (BSSBM) (Abbe et al., 2015), which consists of two communities with $n/2$ nodes each. A pair of nodes within the same community are connected (forming an intra-cluster edge) with probability p, while a pair of nodes in the separate communities are connected with probability q (forming an inter-cluster edge). BSSBMs are quite restricted in terms of their structure, and many more complex SBMs have been developed to model more realistic networks, such as: multiple equal-sized clusters (Symmetric SBMs (SSBM)), allowing two clusters of unequal sizes (Binary Asymmetric SBM (BASBM)) (Hajek et al., 2016b), combining them together and with the existence of outliers (General Structure SBM (GSSBM)) (Hajek et al., 2016a), adding features such as weight labels (Censored SBM (CSBM)) (Hajek et al., 2016b), and by introducing new assumptions such as degree distribution of vertices (Degree Corrected SBM), e.g., (Qin & Rohe, 2013); see (Lee & Wilkinson, 2019) for a survey on SBMs.

A fundamental problem is "recovering" the community structure from a given SBM, and this has spurred a very active area of research. *Exact recovery* is defined as when the probability that the community detection algorithm successfully recovers the ground-truth communities converges to 1 when the size of the input graph (the number of vertices n) goes to infinity, with the probability space is over

[1]Department of Computer Science and Biocomplexity Institute and Initiative, University of Virginia, Virginia, USA. Correspondence to: Dung Nguyen <dungn@virginia.edu>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

the randomness of the SBM process (Abbe et al., 2015). A celebrated result in this area is the exact recovery condition for BSSBM, namely that it is possible if $p = a\log n/n$, $q = b\log n/n$ for constants a and b, and $\sqrt{a}-\sqrt{b} \geq \sqrt{2}$. In some other regimes, for example, when $\sqrt{a}-\sqrt{b} < 2$ in the above setting, or when $p, q = \Theta(1/n)$, it has been shown that exact recovery is impossible. When $\sqrt{a} - \sqrt{b} \geq \sqrt{2}$, exact recovery can be achieved by many different community detection algorithms, such as using a Maximum Likelihood Estimator (MLE), Spectral methods, or by Semi-definite programming (SDP) (Boppana, 1987; McSherry, 2001; Abbe et al., 2015; Massoulié, 2014; Gao et al., 2017; Hajek et al., 2016a; Abbe et al., 2020; Wang et al., 2020). Exact recovery has been studied for more complex SBMs, such as BASBM, GSSBM, and CSBM (Hajek et al., 2016b;a), but the conditions are much more complex. For instance, in BASBM, the threshold for exact recovery is when $\frac{a+b}{2} - \gamma + \frac{(1-2\rho)\tau}{2}\log\frac{\rho(\gamma+(1-2\rho)\tau)}{(1-\rho)(\gamma-(1-2\rho)\tau)} > 1$, where $\tau = (a-b)/(\log a - \log b)$ and $\gamma = \sqrt{(1-2\rho)^2\tau^2 + 4\rho(1-\rho)ab}$ (Hajek et al., 2016b).

Data privacy is a very significant concern in a number of applications. Differential Privacy (DP) (Dwork et al., 2014) has become a *de facto* standard for privacy, due to its rigorous guarantees. DP algorithms guarantee that their outcomes will be similar probabilistically, measured by privacy parameters ϵ, δ, if the input is slightly modified. In context of networks and graph algorithms, two common privacy models have been considered, namely edge- and node-privacy (Kasiviswanathan et al., 2013; Blocki et al., 2013; Mülle et al., 2015; Nguyen et al., 2016; Qin et al., 2017; Imola et al., 2021; Blocki et al., 2013). The edge-privacy model protects the existence and non-existence of an arbitrary edge in the input graph. In contrast, the node-privacy model provides protection to any node and its incident edges. Most work on private algorithms on community structures of graphs has focused on the edge-DP model, since the output contains nodes, e.g., densest subgraph (Nguyen & Vullikanti, 2021), community detection (Hehir et al., 2021; Mohamed et al., 2022; Nguyen et al., 2016). Though the node-privacy model provides a stronger privacy guarantee, the edge-privacy model still provides meaningful protection in a number of applications. A concrete example of the protection of edge-DP is against "Link disclosure" attack in social network analysis (Zhou et al., 2008; Kiranmayi & Maheswari, 2021), where users are modeled as nodes and social relationships between users are edges in the social graphs. For example, in the analysis of a communication graph that models the email interactions between students and faculty members in a university, in which the relationship "who emails whom" is considered sensitive, edge privacy can be applied to protect the sensitive links to be exposed (Jiang et al., 2021). We refer readers to these studies (Li et al., 2023; Jiang et al., 2021) for the motivation for edge-DP and its protection in many practical applications.

Community detection for SBMs under DP constraints (especially edge-DP) has been studied extensively in recent years (Hehir et al., 2021; Mohamed et al., 2022; Seif et al., 2023; Chen et al., 2023; Guo et al., 2023). The first rigorous bound for recoverability in SBMs was established recently by (Mohamed et al., 2022) for the special case of symmetric SBMs; for BSSBMs, they show that the condition for exact recovery is $\sqrt{a} - \sqrt{b} > \sqrt{2}\cdot\sqrt{1+3/2\epsilon}$, and that this can be extended to the case of r communities. *The conditions for exact recovery in all other SBMs under differential privacy remain open.*

1.1. Our contributions

We study the exact recoverability problem in SBMs under the edge DP model— this model is the natural model to consider (instead of node DP), since our goal is to output the community structure. We consider three important extensions to the symmetric SBM model.

- *Binary Asymmetric* with unequal-sized clusters of size ρn and $(1-\rho)n$, for some constant $\rho \in [0, 0.5]$.
- *Binary Censored*, in which edges of a graph $\mathcal{G}(n,p)$ (from the Erdős-Rényi model) are labeled as follows: an intra-cluster (or inter-cluster) edge has label $\mathbf{1}$ (or $\mathbf{-1}$, respectively) with probability $1-\xi$, and has the opposite label $\mathbf{-1}$ (or $\mathbf{1}$, respectively) with probability ξ, for some constant $\xi \in [0, 0.5]$.
- *General Structure*, consisting of multiple and possibly unequal clusters with outlier vertices that do not belong to any cluster. Each intra-cluster connection is generated with probability p, and all other connections are generated with probability q.

We derive the first rigorous recoverability conditions in three extensions of SBMs under the edge DP model, and design polynomial time algorithms for recoverability. Our results significantly extend the prior best theoretical result for recoverability in symmetric SBMs (Mohamed et al., 2022).

We establish the rigorous conditions of parameters for recoverability by sophisticated analysis of the stability of the Semi-definite program estimator on retrieving the true cluster mapping for each SBM variant. The condition involves both the graph model's parameters (edge probabilities p and q or edge label noisiness ξ), and the privacy parameters ϵ and δ, to formalize the roles of different factors in the success of recovery.

The main difficulties are the design and analysis of a set of core conditions named $\mathcal{C}$-concentration for each model

with three simultaneous properties: (1) being satisfied by a graph generated by the SBM with high probability (which refers to probability at least $1 - 1/n^c$ for a constant c, and abbreviated by w.h.p.), (2) persisting under input graph perturbation up to $\log n$ edges, and (3) being sufficient to construct a dual certificate for the SDP Relaxation deterministically. The SBMs of interest are all significantly more complex than the symmetric SBMs, and differ substantially from each other. Each SBM requires a new design of $\mathcal{C}$-concentration, with little common analysis among them.

We summarize the threshold in Table 1. Finally, we design the first polynomial-time algorithms for community detection for SBMs with edge-DP for arbitrary small privacy parameter ϵ, matching the exact recovery threshold in non-private settings when $\epsilon \to \infty$; *none of the prior methods provide these guarantees.*

Our results advance the boundaries of recoverability with privacy on multiple variants of SBM. Our focus here is on the theoretical foundations of the problem. We maintain a full, updated version of this work at (Nguyen & Vullikanti, 2024).

1.2. Related work

(Mohamed et al., 2022) are the first to derive rigorous bounds of exact recovery for SBMs under the edge-DP model. They proposed several approaches, notably applying the Stability mechanism on MLE and SDP estimator to achieve exact recovery in the symmetric settings with arbitrary small ϵ in (ϵ, δ)-DP, with exponential and quasi-polynomial time, respectively. Additionally, their proposed methods can achieve exact recovery in polynomial time but with the cost of $\epsilon = \Omega(\log n)$, or in pure-DP with the cost of exponential time. Their best recovery threshold has the bound $\sqrt{a}-\sqrt{b}$ proportional to $1/\sqrt{\epsilon}$, and matches the non-private setting when $\epsilon \to \infty$. (Seif et al., 2023) improved the computational complexity of (Mohamed et al., 2022) by using i.i.d. vertex sampling with probability ζ and performing the Stability mechanisms on the sampled subgraph. After using private voting via the Laplace mechanism to classify unsampled nodes, they showed that exact recovery is achieved for both sampled and unsampled nodes, with the recovery thresholds increased by a factor of $1/\sqrt{\zeta}$. Their asymptotic running times remain unchanged, being exponential and quasi-polynomial for the MLE- and SDP-based mechanisms, respectively. (Chen et al., 2023) are the first to achieve exact recovery in polynomial time with any constant ϵ. Their method relies on transforming the community detection problem into an optimization problem. They proved that once the optimization is strongly convex, the sensitivity of its solution is bounded, hence adding noises by the Gaussian mechanism provides privacy. This method achieves exact recovery in the BSSBM with the threshold being off from the non-private setting's threshold by some constant, and the bound $\sqrt{a}-\sqrt{b}$ being proportional to $1/\epsilon$ (see Table 1). (Hehir et al., 2021), (Guo et al., 2023), (Ji et al., 2019) studied the community detection in SBMs under DP, but did not focus on the exact recovery questions.

2. Preliminaries

Stochastic Block Models. The stochastic block model (SBM) is a family of random graph models in which a set of vertices $|V| = n$ is partitioned into r clusters (communities): $C_1, \ldots, C_r$ plus some outliers that do not belong to any cluster. In binary forms ($r = 2$), the clusters are represented by a vector $\sigma^* \in \{\pm 1\}^n$ where $\sigma_i^* = 1$ if vertex i belongs to the first cluster and $\sigma_i^* = -1$ otherwise. When $r > 2$, clusters are represented by r binary indicator vectors $\xi_1^*, \ldots, \xi_r^* \in \{0, 1\}^n$ where $\xi_k^*(i) = 1$ if vertex i belongs to the cluster k^{th} and $\xi_k^*(i) = 0$ otherwise. Connections between vertices are generated independently with probability p if the endpoints are in the same cluster and with probability q in other cases to form a set of edges E. In this paper, we focus on the dense regime, where $p = a \log n/n$ and $q = b \log n/n$ for some constants $a \geq b > 0$, of the following three variants of SBMs:

1. *Binary Asymmetric (BASBM)*: An SBM with $r = 2$ in which the first cluster contains $\lfloor n\rho \rfloor$ and the second one contains $\lceil n(1 - \rho) \rceil$ vertices for some constant $\rho \in [0, 0.5]$.

2. *Binary Censored (BCSBM)*: An SBM with $r = 2$ and $p = q$. In other words, edges are generated by an Erdos-Renyi model $\mathcal{G}(n, p)$. Each edge (i, j) has label $L_{ij} \in \{\pm 1\}$ independently drawn from the distribution: $P_{L_{ij}} = (1 - \xi)\mathbf{1}_{L_{ij} = \sigma_i^* \sigma_j^*} + \xi \mathbf{1}_{L_{ij} = -\sigma_i^* \sigma_j^*}$.

3. *General Structure (GSSBM)*: An SBM with $r > 2$, where the k^{th} cluster C_k has size $K_k = \rho_k n$ and $\rho_1 \geq \ldots \geq \rho_r > 0$; and $n - \sum_{k \in [r]} K_k$ outliers. We use $k = 0$ (e.g., in C_0, K_0) to refer to the outliers, but the edges among them are not considered intra-cluster.

Exact recovery. Given the input graph $G = (V, E)$ as described above and an algorithm $\mathcal{A}$, *exact recovery* means that $\mathcal{A}$ outputs the ground-truth cluster vector σ^* up to a permutation w.h.p., i.e., with probability tending to 1 when n goes to infinity: $\Pr[\mathcal{A}(G) \neq \pm \sigma^*] = o(1)$. For $r > 2$, we often use the cluster matrix: $\Pr[\mathcal{A}(G) \neq Z^*] = o(1)$, where $Z^* = \sum_{k \in [r]} \xi_k^* (\xi_k^*)^T$.

Exact recovery is not always possible, for example, in the *sparse regime* $p, q = \Omega(1/n)$. In the symmetric SBM, exact recovery is possible in the *dense regime* if and only $\sqrt{a} - \sqrt{b} \geq \sqrt{r}$. For Binary Asymmetric SBM, the threshold for exact recovery is when $\frac{a+b}{2} -$

Algorithm	SBM Model	Recovery threshold	Running time
Non-private SDP	Binary Symmetric	$\sqrt{a}-\sqrt{b}\geq\sqrt{2}$	$O(poly(n))$
MLE-Stability[(1)]	Binary Symmetric	$\sqrt{a}-\sqrt{b}\geq\sqrt{2}\sqrt{1+3/(2\epsilon)}$	$O(exp(n))$
SDP-Stability[(1)]	Binary Symmetric	$\sqrt{a}-\sqrt{b}\geq\sqrt{2}\sqrt{2+3/(2\epsilon)}$	$n^{O(\log n)}$
RR + SDP[(1)]	Binary Symmetric	$\epsilon=\Omega(\log n)$, $\sqrt{a}-\sqrt{b}>\sqrt{2}\times\frac{\sqrt{e^\epsilon+1}}{\sqrt{e^\epsilon-1}}+\frac{1}{\sqrt{e^\epsilon-1}}$	$O(poly(n))$
(Chen et al., 2023)	Binary Symmetric	$\sqrt{a}-\sqrt{b}\geq 16, a-b\geq\frac{500^2}{\epsilon^2}+\frac{64}{\epsilon}$ **(4)**	$O(poly(n))$
(Seif et al., 2023) (MLE)	Binary Symmetric	$\sqrt{a}-\sqrt{b}\geq\sqrt{2/\zeta}\sqrt{1+3/(2\epsilon)\Theta(\log(a/b))}$	$O(exp(n))$
(Seif et al., 2023) (SDP)	Binary Symmetric	$\sqrt{a}-\sqrt{b}\geq\sqrt{2/\zeta}\sqrt{1+3/(\epsilon)\Theta(\log(a/b))}$ **(5)**	$n^{O(\log n)}$
$\mathcal{M}^{SDP}_{\text{Stbl Fast}}$ **(Ours)**	Binary Symmetric[(2)]	$\sqrt{a}-\sqrt{b}\geq\sqrt{2}\sqrt{1+2/\epsilon}$ (Cor. 1) $\sqrt{a}-\sqrt{b(1+\log\frac{a}{b})}>\sqrt{c\log\frac{a}{b}/\epsilon}$ **(6)**	$O(poly(n))$
$\mathcal{M}^{SDP}_{\text{Stbl Fast}}$ **(Ours)**	Binary Asymmetric	Theorem 1	$O(poly(n))$
	Binary Censored	Theorem 2	$O(poly(n))$[(3)]
	General Structure	Theorem 3	$O(poly(n))$[(3)]

Table 1. Summary of the algorithms; SBM variants in the dense regime $p=a\log n/n, q=b\log n/n$; recovery thresholds on a, b and ϵ; time-complexity; at $\delta=n^{-2}$. For ϵ-DP, both (Mohamed et al., 2022; Chen et al., 2023) achieve exact recovery in exponential time. (1) Algorithms of (Mohamed et al., 2022). (2) The threshold derives from our BASBM with $\rho=1/2$ (Corollary 1). (3) for Binary Censored and General Structure, $\mathcal{M}^{SDP}_{\text{Stbl Fast}}$ takes $O(poly(n))$ if parameters a, b and ξ are known by the algorithms, otherwise we use $\mathcal{M}^{SDP}_{\text{Stbl}}$ that takes $n^{O(\log n)}$. (4) is loosely equivalent to $\sqrt{a}-\sqrt{b}\geq 500/\epsilon+8/\sqrt{\epsilon}$. (5)$\zeta$ denotes vertex sampling probability. (6) when $\epsilon\to\infty$, this condition is roughly equivalent to $a>b$.

$\gamma+\frac{(1-2\rho)\tau}{2}\log\frac{\rho(\gamma+(1-2\rho)\tau)}{(1-\rho)(\gamma-(1-2\rho)\tau)}>1$, where $\tau=(a-b)/(\log a-\log b)$ and $\gamma=\sqrt{(1-2\rho)^2\tau^2+4\rho(1-\rho)ab}$. In the case of the Binary Censored model, the threshold is $a(\sqrt{1-\xi}-\sqrt{\xi})^2>1$. The condition for exact recovery in the General Structure is stated in (Hajek et al., 2016b).

Privacy model. We consider the exact recovery problem under the edge-privacy model (Karwa et al., 2011). A randomized algorithm $\mathcal{A}$ is (ϵ,δ)-differentially private (DP) if for any pair of neighbor graphs that differ by exact one edge (denoted as $G\sim G'$), $\Pr[\mathcal{A}(G)\in S]\leq e^\epsilon\Pr[A(G')\in S]+\delta$ for any $S\subseteq Range(A)$.

Stability mechanism. Given a function f, an input graph G is called c-stable if $\forall G': dist(G,G')\leq c$, we have $f(G)=f(G')$, where $dist$ denotes the Hamming distance. A graph is unstable if it is 0-stable. The *distance to instability* of input G on f, denoted by $d_f(G)$ is the shortest distance to reach to an unstable graph G' from G. The f-based Stability mechanism, as defined in Algorithm 1, is (ϵ,δ)-DP (Dwork et al., 2014; Mohamed et al., 2022) (The formal statement and proof are presented in Theorem 5). It first calculate G's private distance to instability, by adding a Laplacian noise with magnitude $1/\epsilon$ (Lap(b) denotes the Laplace distribution with PDF $\text{Lap}(x|b)=1/(2b)\exp(-|x|/b)$), as the distance to instability always have sensitivity of 1. Line 5 of the algorithm returns an undefined output ($\perp$) which keeps the privacy analysis simple, but in practice can be replaced by returning a random output.

Utility of Stability mechanism. The utility of the Stability mechanism dictates its ability to achieve exact recovery. Let f be any community detection algorithm that outputs the communities given an input G. Lemma 1 states that with appropriate selections of δ, if an input graph G is $O(\log n)$-stable under f, then w.h.p. (with probability at least $1-n^{-\Omega(1)}$), the mechanism returns $f(G)$. We extensively utilize this property in our analyses. Hereafter, for simplicity, we will refer to $O(\log n)$-stable (or distance up to $O(\log n)$) as $\log n$-stable (or distance up to $\log n$).

Algorithm 1 $\mathcal{M}^f_{\text{Stbl}}(G)$: Stability Mechanism

1: $\tilde{d}_f(G)\leftarrow d_f(G)+\text{Lap}(1/\epsilon)$
2: **if** $\tilde{d}_f(G)>\frac{\log 1/\delta}{\epsilon}$ **then**
3: Output $f(G)$
4: **else**
5: Output $\perp$
6: **end if**

Lemma 1. *(Full proof in Lemma 6) The f-based Stability mechanism with $\delta=n^{-c}$ has $\Pr[M^f(G)\neq f(G)]\leq n^{-k_1}+n^{-k_2}$, if a graph G is $\frac{c+k_1}{\epsilon}\log n$-stable under function f with probability at least $1-n^{-k_2}$. When $k_1,k_2=\Omega(1)$, $\Pr[\mathcal{M}^f_{\text{Stbl}}\neq f(G)]\leq n^{-\Omega(1)}$.*

Problem statement. Given an input graph $G=(E,V)$, with $|V|=n$, assumed to generated by an SBM with a fixed but unknown cluster vector σ^* (or community matrix Y^* in case of GSSBM), design an algorithm $\mathcal{A}$ such that: (1) $\Pr[\mathcal{A}(G)\neq\pm\sigma^*]=o(1)$ (exact recovery property) and (2) $\mathcal{A}$ is (ϵ,δ)-DP.

3. Exact Recovery under Differential Privacy

In non-private settings, using Semidefinite programming (SDP) to estimate the clusters is an established method for community detection and exact recovery. We apply the SDP estimator with the Stability mechanism to design our algorithms for differentially private exact recovery.

In this Section, we will present an inefficient algorithm by a direct application of the Stability mechanism to output a private SDP estimator with privacy. We show in this section a high-level overview of our analyses by introducing the generic conditions and analysis routines to guarantee exact recovery for all three targeted SBMs above. In Section 4, we will present the custom-built analyses for each of the three models to fulfill the proof. Finally, in Section 5, we will show how to design a novel polynomial-time algorithm that still retains all privacy and utility guarantees. Due to space limits, we leave the complete proofs in the Appendix.

There are two main steps in designing the algorithms. The first step is to define the SDP relaxation of the SBMs based on the input graph's adjacency matrix A. The second step is to treat the optimal solution of the SDP as a function (denoted as $SDP(G)$) and apply it in the role of f in Algorithm 1. Here is the SDP for the BASBM model:

$$\max_{\sigma} \sum_{ij} A_{ij}\sigma_i\sigma_j \quad (1) \qquad \text{s.t. } \sigma_i \in \{\pm 1\}, i \in [n] \qquad o^T\mathbf{1} = n(2\rho - 1)$$

$$\widehat{Y}_{SDP} = \arg\max_Y \langle A, Y\rangle \quad (2) \qquad \text{s.t. } Y \succcurlyeq 0 \qquad Y_{ii} = 1, i \in [n] \qquad \langle J, Y\rangle = (n(2\rho-1))^2$$

Let σ be our estimator vector with $\sigma_i = 1$ if i is in the first cluster and $\sigma_i = -1$ otherwise. The optimization problem (1) maximizes the differences between intra-cluster edges and inter-cluster edges while fixing the size of the two clusters (the last constraint), namely Maximum Likelihood Estimator (MLE). Solving this is NP-hard. We relax the problem into the SDP optimization (2) as follows: let $Y = \sigma\sigma^T$, $\sigma = \pm 1$ will be transformed to $Y_{ii} = 1$, and the size constraint will be equivalent to $\langle J, Y\rangle = (n(2\rho-1))^2$. All feasible matrices are rank-one positive semi-definite, which is then relaxed to $Y \succcurlyeq 0$. Depending on the unique specifications of each SBM, their equivalent SDPs are vastly different. Further details are presented in (Hajek et al., 2016b).

Applying the Stability mechanism. After defining the SDP (and the function $SDP(G)$, which denotes the optimal solution), we can directly apply it to the Stability mechanism by substituting $f(G)$ by $SDP(G)$ to form an SDP-based Stability mechanism. We note that for any function f, the $\mathcal{M}^f_{\text{Stbl}}$ is (ϵ, δ)-DP. Theorem 5 formalizes the privacy analysis of the mechanism. *Exact recovery* requires extensive analysis of SDP under the effect of the mechanism. To achieve exact recovery, $SDP(G)$ must be $\log n$-stable w.h.p., due to Lemma 1. Though (Mohamed et al., 2022) shows the analysis for the symmetric SBMs, and it cannot be extended easily to cover any of the three models above, due to their different SDP formulas, as well as different criteria of the certificates of the optimality of the SDPs. We next present the generic routine, which consists of the three main steps below, to prove the $\log n$-stability property for a generic SBM, and then provide a specific procedure for each of them in Section 4.

Generic analysis of $\log n$-stability of $SDP(G)$ is centered around the concept of $\mathcal{C}$-concentration, parameterized by a tuple of constants $\mathcal{C}$. It is a set of special conditions constructed from the specifications of the assumed SBM. For example, $\|A - \mathbb{E}[A]\| \leq c\sqrt{\log n}$ is a condition parameterized by a constant c, where expected value $\mathbb{E}[A]$ of the adjacency matrix A is dictated by the SBM. A graph G is called $\mathcal{C}$-concentrated when it satisfies all conditions under $\mathcal{C}$. Each SBM, and its equivalent SDP, has a distinct $\mathcal{C}$-concentration. To prove that $SDP(G)$ is $\log n$-stable, we have to complete the following steps:

Step 1. $\mathcal{C}$-concentration w.h.p. Over the randomness of the generation process of an SBM, we prove that a random graph G is $\mathcal{C}$-concentrated w.h.p.. It often requires each condition being satisfied w.h.p. and taking union bound on all conditions. For example, we prove that the condition $\|A - \mathbb{E}[A]\| \leq c\sqrt{\log n}$ be satisfied w.h.p. for a graph generated by the SBM. This step also sets the restrictions on the constants $\mathcal{C}$, e.g., c in the above condition must be a positive constant, which will determine the thresholds for the exact recovery. It is formalized by the following proposition:

Proposition 1. *Given graph G generated by an SBM of specific settings and its respective $\mathcal{C}$-concentration conditions, there exists some tuples of constants $\mathcal{C} = \{c_1, c_2, \ldots\}$ such that G is $\mathcal{C}$-concentrated with probability at least $1 - n^{-\Omega(1)}$.*

Step 2. $\mathcal{C}$-concentration persists under up to $\log n$ edges modifying. By proving the following proposition, we show that modifying $\log n$ edges of a $\mathcal{C}$-concentrated graph G does not destroy its concentration properties (up to a new tuple $\mathcal{C}'$ as long as $\mathcal{C}'$ satisfies the restriction in Step 1). For example, with the condition $\|A - \mathbb{E}[A]\| \leq c\sqrt{\log n}$, modifying up to $\log n$ edges of the input graph G flips at most $2\log n$ bits of A (due to the symmetry of A). For the condition to persist, we need a new constant $c' = c + 2$. Since $c' > 0$, it is a valid constant of $\mathcal{C}$.

Proposition 2. *If G is $\mathcal{C}$-concentrated then for every graph $G' : d(G, G') < c\log n$, i.e., G' can be constructed by flipping at most $c\log n$ connections of G, G' is $\mathcal{C}'$-concentrated, where $\mathcal{C}'$ is a valid tuple of constants depending only on $\mathcal{C}$, c, and the SBM's constant parameters.*

Step 3. $\mathcal{C}$-concentration implies SDP optimality at the ground-truth. In this step, we will construct a (deterministic) dual certificate for the SDP Relaxation using the $\mathcal{C}$-concentration's conditions. Once we do that, we can confirm that the ground-truth cluster matrix (Y^* or Z^*) is the unique optimal solution of the SDP Relaxation. In other words, when $\mathcal{C}$-concentration holds, $SDP(G)$ always outputs the ground-truth clusters. Lemma 2 states the certificates for the BASBM, i.e., the ground-truth community matrix Y^* is the unique and optimal solution of the SDP given appropriate D^* and S^*. (Hajek et al., 2016b) shows that D^* and S^* exist w.h.p. for the non-private setting. In our design, we show that from $\mathcal{C}$-concentration's conditions (such as $\|A-\mathbb{E}[A]\| \leq c\sqrt{\log n}$), we can always, i.e., with probability 1, construct D^* and S^* that satisfies Lemma 2. Applying the lemma, we show that $\mathcal{C}$-concentration implies that the SDP has the optimal solution at the exact ground-truth.

Lemma 2. *(Lemma 3 of (Hajek et al., 2016b)) Suppose there exist $D^* = diag\{d_i^*\}$ and $\lambda^* \in \mathbb{R}$ such that $S^* = D^* - A + \lambda^* J$ satisfies $S^* \succcurlyeq 0, \lambda_2(S^*) > 0$ and $S^*\sigma^* = 0$. Then Y^* is the unique solution of the program $SDP(G)$.*

Proposition 3. *If G is $\mathcal{C}$-concentrated under an SBM, then $SDP(G) = Y^*$(or Z^*), i.e, the optimal solution $SDP(G)$ is the ground-truth cluster matrix Y^* (or Z^*).*

Exact recovery. Step 2 & 3 guarantee that when an input graph G is $\mathcal{C}$-concentrated under an SBM with specific settings, the function $SDP(G)$ is $\log n$-stable. Since G is $\mathcal{C}$-concentrated, $SDP(G)$ outputs the ground-truth Y^*, or $SDP(G) = Y^*$, by Proposition 3. Any graph G' created by flipping up to $\log n$ edges of G is $\mathcal{C}'$-concentrated, by Proposition 2. Now, applying Proposition 3 for G', we have $SDP(G) = SDP(G') = Y^*$. In other words, G is $\log n$-stable under the function SDP.

Finally, Step 1 shows that $\mathcal{C}$-concentration happens w.h.p. for an arbitrary graph generated by the SBM. In other words, if G is generated by an SBM with appropriate parameters, G is $\log n$-stable under the function SDP w.h.p.. Applying Lemma 1, substituting f by SDP, $\Pr[\mathcal{M}_{\text{Stbl}}^{SDP}(G) \neq Y^*(\text{or } Z^*)] \leq n^{-\Omega(1)}$, or the ground-truth clusters are recovered w.h.p..

4. Stability analyses for SBMs

In this Section, we develop the specific analysis for each of the three SBMs based on the generic routine in Section 3.

4.1. Binary Asymmetric

The SDP Relaxation of the Binary Asymmetric model is shown in Section 3. Recall that in BASBM, two clusters have size ρn and $(1-\rho)n$, $p = a\log n/n$, $b = b\log n/n$. We define the $\mathcal{C}$-concentration as follows:

Definition 1. *G is called $\mathcal{C} = (c_1, c_2, c_3, c_4)$-concentrated, in which $c_i > 0$, if G satisfies all 4 conditions:*

- $\|A - \mathbb{E}[A]\|_2 \leq c_1\sqrt{\log n}$
- $\check{x}^\top D^* \check{x} + \mathcal{J}(\check{x}) > c_2 \log n$
- $\|(D^* - \mathbb{E}[D^*])\check{x}\|_2 \leq c_3\sqrt{\log n}$
- $d_i^* \geq c_4 \log n$ *for every* $i \in [n]$;

in which $\tau = \frac{a-b}{\log a - \log b}$, $\lambda^* = \tau \log n/n$, $d^* \in \mathbb{R}^n : d_i^* = \sum_{j=1}^n A_{ij}\sigma_i^*\sigma_j^* - \lambda^*(2K-n)\sigma_i^*$, $D^* = diag\{d^*\}$, I *is the identity matrix,* J *is the all-one matrix,* $\mathcal{J}(x) = \left(\lambda^* - \frac{p+q}{2}\right)x^\top Jx$, $\check{x} = \arg\max_x x^\top Jx$ *subject to* $\|x\|_2 = 1$ *and* $\langle x, \sigma^*\rangle = 0$, $\tilde{h}(x) = a\rho + b(1-\rho) - \sqrt{(\tau(1-2\rho)-x)^2 + 4\rho(1-\rho)ab} + \frac{\tau(1-2\rho)-x}{2}\log\frac{\rho b}{(1-\rho)a}$.

Assumptions of the parameters. These assumptions will be used in our analyses to derive the conditions in which exact recovery under DP is feasible.

The exists a constant $c > 0$ that for $\tilde{h}(x)$ defined above:

$$\tilde{h}(c) = 1 + \Omega(1) \tag{3}$$

Lemma 3. *(Complete proof in Lemma 14) A graph G generated by a Binary Asymmetric SBM for some constant $\rho \in [0, 0.5]$ and with $p = \frac{a\log n}{n}$ and $q = \frac{b\log n}{n}$, there exists some tuples of constants $\mathcal{C}$ that G is $\mathcal{C}$-concentrated with probability at least $1 - n^{-\Omega(1)}$.*

Proof sketch. The adjacency matrix A is a random matrix where each cell is drawn i.i.d with probability of being 1 is $\log n/n$. The first condition follows directly from Lemma 7, saying with these properties, A is usually not far from its expectation $\mathbb{E}[A]$. The second and third conditions are highly complex, and we have to decompose them into sub-components for further analysis. Notice that D^*, by its definition, is a random matrix computed from A. Therefore we can treat $\check{x}^T D^* \check{x}$ and $\|(D^* - \mathbb{E}[D^*])\check{x}\|$ as functions that take independent variables as input, and apply Talagrand inequality (Lemma 8) that says the output of 1-Lipschitz convex functions is not far from its expected values. They are then transformed to expressions of $\log n$ or $\sqrt{\log n}$ as we see in the r.h.s. of the conditions. With that, the remaining things to do is to prove the functions of interest are (1) convex and (2) Liptschitz continuous. The fourth condition involves d_i^*, or more specifically, $\sum_j A_{ij}\sigma_i^*\sigma_j^*$. This quantity is equal to the difference between two Binomial distributions with different numbers of trials and probabilities of success, and can be analyzed by Lemma 9. The final detail of the proof is the tail bound of Lemma 9 refers to the assumption in Equation 3, so that the tail probability is set to less than $n^{-1-\Omega(1)}$ for each d_i^*.

Lemma 4. *(Complete proof in Lemma 15) If G is $\mathcal{C} = (c_1, c_2, c_3, c_4)$-concentrated then for every graph G' : $d(G, G') < \frac{c}{\epsilon}\log n$, i.e., G' can be constructed by flipping at most $\frac{c}{\epsilon}\log n$ connections of G, G' is $\mathcal{C}' = (c_1', c_2', c_3', c_4')$-concentrated, where $c_1' = c_1 + \sqrt{2c/\epsilon}, c_2' = c_2 - c/\epsilon, c_3' = c_3 + \sqrt{2c(1-\rho)/\epsilon\rho}, c_4' = c_4 - c/\epsilon$.*

Proof sketch. The general idea is to construct worst-case scenarios of quantities in the l.h.s. of the conditions when adding/removing $\log n$ edges and edit $\mathcal{C}$ accordingly such that all four conditions still hold for the new graph. For example, we construct the adjacency matrix A' obtained by modifying $\log n$ entries of A. Assume $\mathbb{E}[A] = \mathbb{E}[A']$ because they are assumed to be generated by the same SBM, we can apply the triangle inequality property of ℓ-norm to prove that the l.h.s. will be increased by at most $\Theta\sqrt{\log n}$. We then adjust c' by the same constant to complete the proof for the first condition. For the second condition, notice that D^* will change by edge modifying. Let $\Delta = D'^* - D^*$, with D'^* being the same quantity as D^* but in G'. We can prove that Δ is also diagonal, with the sum of its entries being at most $\Theta(\log n)$, and its largest eigenvalue is $\Theta(\log n)$. Then the difference after modifying $\log n$ edges is reduced to the quantity $\check{x}^T\Delta\check{x} \le \lambda_{max}(\Delta)\|\check{x}\|_2^2 \le \Theta(\log n)$, which means we only need to alter c_2 by some small constant for the condition to persist. For the third condition, we reduce the change in the l.h.s. of the condition by triangle inequality to $\|(D'^* - D^*)\check{x}\|$. Expand this quantity by expanding D^* and D'^* by their definitions to expressions of A_{ij} and A'_{ij}, we can reduce it to the F-norm of $A - A'$, which is no greater than $\Theta(\log n)$). Since the r.h.s. of the third condition is also $\Theta(\sqrt{\log n})$, we only need to change c_3 by some small constant for it to hold. For the fourth condition, for each i, the difference quantity is bounded by $\sum_j (A'_{ij} - A_{ij})\sigma_i^*\sigma_j^*$. Because A and A' differ by at most $\Theta(\log n)$ entries, it is clear that changing the constant c_4 a bit will cover the changes.

Lemma 5. *(Complete proof in Lemma 17) If G is $\mathcal{C}$-concentrated, then $SDP(G) = Y^*$, i.e., the SDP's optimal solution is the ground-truth cluster matrix $Y^* = \sigma^*\sigma^{*T}$.*

Proof sketch. We will construct a (deterministic) dual certificate by the conditions of $\mathcal{C}$-concentration. Lemma 16 states that $SDP(G) = Y^*$, if we can construct a matrix $S^* \overset{def}{=} D^* - A + \lambda^* J$ satisfies (1) $S^* \succcurlyeq 0$, (2) $\lambda_2(S^*) > 0$, and (3) $S^*\sigma^* = 0$. The condition (3) is easy to verify by expanding the definition of S^*. Because of this, proving $\inf_{x\perp\sigma^*, \|x\|=1} x^T S^* x > 0$ is sufficient to satisfy all remaining conditions (1) and (2), since all feasible x plus σ^* will include a basis for the whole space, which means $\forall y : y^T S^* y \ge 0$ ($S^* \succcurlyeq 0$) and the solution set of $S^* y = 0$ has only 1 dimension ($\lambda_2(S^*) > 0$).

To prove $x^T S^* x > 0$, we have to utilize all four conditions of $\mathcal{C}$-concentration. First we expand $x^T S^* x = x^T D^* x - x^T A x + x^T \lambda^* J x$. Adding $x^T\mathbb{E}[A]x - x^T\mathbb{E}[A]x$ to the r.h.s. and grouping $x^T A x - x^T\mathbb{E}[A]x$, we can use the **first condition** to bound this quantity by $c_1\sqrt{\log n}$. We next expand the remaining $\mathbb{E}[A] = \frac{p-q}{2}Y^* + \frac{p+q}{2}J - pI$, and reduce the equation to $x^T S^* x \ge p - c_1\sqrt{\log n} + x^T D^* x + (\lambda^* - \frac{p+q}{2})x^T J x$.

Let $t(x) = x^T D^* x + (\lambda^* - \frac{p+q}{2})x^T J x$. We define $E = span(\check{x}, \sigma^*)$. Any $y : y \perp \sigma^*, \|y\|_2 = 1$ can be represented as $y = \beta\check{x} + \sqrt{1-\beta^2}x$ for $x \in \{x : x \perp E, \|x\|_2 = 1\}$ and $\beta \in [0, 1]$. For all $x : x \perp \sigma^*, \|x\|_2 = 1$:

$\inf_x t(x) = \inf_{x,\beta\in[0,1]} t(\beta\check{x} + \sqrt{1-\beta^2}x)$:

$$
\begin{aligned}
&\ge \inf_{\beta\in 0,1} (\beta^2(\check{x}^T D^* \check{x} + \mathcal{J}(\check{x})) \\
&+ \inf_{x,\beta\in[0,1]} (2\beta\sqrt{1-\beta^2}x^T D^* \check{x} + (1-\beta^2)x^T D^* x) \\
&\ge \inf_{\beta\in 0,1} (\beta^2(\check{x}^T D^* \check{x} + \mathcal{J}(\check{x})) \\
&\quad + (1-\beta^2)c_4 \log n) - c_3\sqrt{\log n} \\
&\ge \frac{1}{2}\min\{c_2, c_4\}\log n - c_3\sqrt{\log n}, . \qquad (4)
\end{aligned}
$$

where in the second last inequality, we apply the **second condition, third condition, fourth condition** to bound $\check{x}^\top D^*\check{x} + \mathcal{J}(\check{x})$, $\|(D^* - \mathbb{E}[D^*])\check{x}\|_2$, $x^T D^* x$, respectively. Applying this result to $x^T S^* x \ge p - c_1\sqrt{\log n} + x^T D^* x + (\lambda^* - \frac{p+q}{2})x^T J x \ge \frac{1}{2}\min\{c_2, c_4\}\log n - (c_1 + c_3)\sqrt{\log n} + p > 0$, when n is large enough and the Theorem follows.

Finally, by the arguments in Section 3, we show that $\mathcal{M}_{\text{Stbl}}^{SDP}$ achieves exact recovery as follows:

Theorem 1. *(Complete proof in Theorem 6)*

Given a graph G generated by a Binary Asymmetric SBM with two communities sized ρn and $(1-\rho)n$ for some constant ρ, and with $p = \frac{a\log n}{n}$ and $q = \frac{b\log n}{n}$, $\sqrt{a} - \sqrt{b(1+\log\frac{a}{b})} > \sqrt{c\log\frac{a}{b}/\epsilon}$, and $\tilde{h}(c/\epsilon) > 1$, $\mathcal{M}_{\text{Stbl}}^{SDP}$ with $\delta = n^{-c}$ exactly recovers the ground-truth community σ^*, i.e., $\Pr[\mathcal{M}_{\text{Stbl}}^{SDP}(G) \ne Y^*] = n^{-\Omega(1)}$.

Corollary 1. *When $\rho = 1/2$, the condition for exact recovery is $\sqrt{a} - \sqrt{b} \ge \sqrt{2}\sqrt{1 + c/\epsilon}$ and $\sqrt{a} - \sqrt{b(1+\log\frac{a}{b})} > \sqrt{c\log\frac{a}{b}/\epsilon}$.*

4.2. Censored Binary

Privacy model. The adjacency matrix $A(G)$ is defined as $A_{ij} = 0$ if there is no edge between i and j. $A_{ij} = L_{ij}$ if there is an edge generated between i and j. In this section, we define the neighborhood between two graphs $G \sim G'$ if $A(G)$ and $A(G')$ differ by exactly two entries (due to the symmetry). This privacy model can protect the existence (and the non-existence) of an arbitrary edge (i, j), where

any two neighboring adjacency matrices differ at element ij (and ji): $A(G)_{ij} = 0$ (not an edge) and $A(G')_{ij} \neq 0$ (an edge). It can also protect the label of an arbitrary edge (i, j) whenever (i, j) exists in the input graphs, that any two neighboring adjacency matrix differ as follows: $A(G)_{ij} = -1$ and $A(G')_{ij} = +1$.

SDP Relaxation. We reuse the notation and arguments for the Binary Asymmetric model to form the following optimization problems for the Censored Binary model. Even though they do not look much different from the BASBM's SDP, they have vastly different characteristics, because the adjacency matrices A, in this case, implies much more topological information.

$$\max_{\sigma} \sum_{ij} A_{ij}\sigma_i\sigma_j \quad (5) \qquad \text{s.t. } \sigma_i \in \{\pm 1\}, i \in [n]$$

$$\widehat{Y}_{SDP} = \arg\max_Y \langle A, Y\rangle \quad (6) \qquad \text{s.t. } Y \succcurlyeq 0,\ Y_{ii} = 1, i \in [n]$$

Assumptions of parameters. Let $p = a\log n/n$ for some fixed constant a (in the random edge generation model $G(n, p)$). For the random label model: $h(\xi, a) = a(\sqrt{1-\xi} - \sqrt{\xi})^2 > 1$, or $h(\xi, a) = 1 + \Omega(1)$.

Definition 2. *G is called $\mathcal{C}$-concentrated if there exists a tuple $\mathcal{C} = (c_1, c_2)$ such that G satisfies two conditions:*

- $\|A - \mathbb{E}[A]\|_2 \leq c_1\sqrt{\log n}$
- $d_i^* \geq c_2 \log n, i \in [n]$

where $d_i^* = \sum_{j=1}^n A_{ij}\sigma_i^*\sigma_j^*$ for every $i \in [n]$.

Lemma 22, 23, and 25 are the analogues of Lemma 3, 4, and 5 for BASBM. The first condition can be proved w.h.p. using the same strategy as in BASBM. The second condition, however, cannot be easily adapted. We can transform $d_i^* = \sum_{j=1}^{n-1} X_j$, where $X_j \overset{i.i.d.}{\sim} p(1-\xi)\beta_{+1} + p\xi\beta_{-1} + (1-p)\beta_0$, in which β_x is the Dirac delta function at x. In this case, d_i^* cannot be represented by the difference between two Binomial distributions as in BASBM, and we have to utilize Chernoff bound: $\Pr[\sum_{j=1}^n X_j < c_2 \log n] \leq \exp(-n\ell(\frac{c_2\log n}{n}))$ where the function $\ell(x)$ is defined as $\ell(x) = \sup_{\lambda\geq 0} -\lambda x - \log \mathbb{E}[e^{-\lambda X}]$. We then solve the supremum at λ^* and substituting x by $c_2 \log n/n$, and utilizing the assumption $h(\xi, a) > 1$ to determine the tail probability, which is bounded by $n^{-1-\Omega(1)}$.

Theorem 2. *(Complete proof in Theorem 7) Given graph G generated by a CBSBM as described above where $c/\epsilon < a$ and $h(\xi, a) > 1$, M_{Stbl}^{SDP} with $\delta = n^{-c}$ exactly recovers the ground-truth community Y^*, i.e., $\Pr[M_{\text{Stbl}}^{SDP}(G) \neq Y^*] = n^{-\Omega(1)}$*

4.3. General Structure

SDP Relaxation. Unlike the two other models, we use a set of indicator vectors ξ_k to map a vertex to its cluster: $\xi_k(i) = 1$ if $i \in C_k$ and $\xi_k(i) = 0$ otherwise (ξ_k refers to the variables while ξ_k^* is the ground-truth). We form the following optimization problems: the MLE on the left and the transformed SDP Relaxation on the right.

$$\max_{\xi} \sum_{ij} A_{ij} \sum_{k\in[r]} \xi_k(i)\xi_k(j) \quad \text{s.t. } \xi_k \in \{0,1\}^n, k \in [r];\ \xi_k^T\mathbf{1} = K_k, k \in [r];\ \xi_k^T\xi_{k'} = 0, k \neq k' \quad (7)$$

$$\widehat{Z}_{SDP} = \arg\max_Y \langle A, Z\rangle \quad \text{s.t. } Z \succcurlyeq 0,\ Z_{ii} \leq 1, i \in [n],\ Z_{ij} \geq 0, i, j \in [n],\ \langle I, Z\rangle = \sum_{k\in[r]} K_i,\ \langle J, Z\rangle = \sum_{k\in[r]} K_i^2 \quad (8)$$

Due to the space limits, we only present the $\mathcal{C}$-concentration definition for GSSBM and the final exact recovery statement and its conditions (Theorem 3) here. We show the complete analysis in Section E.

Definition 3. *G is called $\mathcal{C}$-concentrated if there exists a tuple of constants $\mathcal{C} = (c_1, c_2, c_3, c_4, c_5)$ such that G satisfies these conditions:*

- $\|A(G) - \mathbb{E}[A(G)]\| \leq c_1\sqrt{\log n}$
- $\min_{i\in[n]} s_i \geq (b + 2c_2)\rho_{k(i)} \log n$
- $\max_{i\in[n],k:k\neq k(i)} e(i, C_k) \leq (b + c_2)K_k \log n/n - c_3 \log n$
- $\min_{i,j:k(i)k(j)[k(i)-k(j)]\neq 0} e(C_{k(i)}, C_{k(j)}) \geq K_{k(i)}K_{k(j)}q - 2\sqrt{K_{k(i)}K_{k(j)}}\sqrt{\log n} - c_4 \log n$
- $\max_{i\in C_0} e(i, C_{k:k\neq 0}) \geq \tilde{\tau}K_r\frac{\log n}{n} - c_5 \log n$,

where $k(i)$ is i's cluster, $e(i, C_k)$ is the number of edges between a node i and nodes from cluster C_k, $e(C_k, C_{k'}) = \sum_{i\in C_k} e(i, C_{k'})$, $s_i = e(i, C_{k(i)})$, $\tilde{\tau} = b + 2c_2$.

Theorem 3. *(Complete proof in Theorem 8) Let $I(x, y) = x - y\log\frac{ex}{y}$. Given a graph G generated by a GSSBM as above, M_{Stbl}^{SDP} with $\delta = n^{-c}$ exactly recovers the ground-truth community Z^*, i.e., $\Pr[M_{\text{Stbl}}^{SDP}(G) \neq Z^*] = n^{-\Omega(1)}$, if the following conditions are satisfied: $I(a, b + \frac{2c}{\epsilon\rho_{min}}) > 1/\rho_{min}$; $I(b, b + \frac{c}{\epsilon}(\frac{1}{\rho_{min}} - 1)) > 1/\rho_{min}$; $I(b, b + \frac{c}{\epsilon}(\frac{2}{\rho_{min}} - 1)) > 1/\rho_{min}$.*

5. Polynomial-time algorithm

In Algorithm 1, calculating $d_{SDP}(G)$ will takes at least $n^{O(\log n)}$ times due to calculating $d(G)$. The main idea is if we can estimate $d(G)$ faster, we can design a faster algorithm. Observe that when G is $\mathcal{C}$-concentrated, $d(G) \geq c\log n/\epsilon$. We test if the input graph is $\mathcal{C}$-concentrated and if the graph pass the test, we can set $\widehat{d}(G) = d(G) \geq c\log n/\epsilon$ and use $\widehat{d}$ instead of d. If G fails the test, we

compute $\widehat{d}(G) = \min(d(G), c\log n/\epsilon)$. The main challenge is that, testing $\mathcal{C}$-concentration requires knowledge of the SBMs, i.e., p, q and **most importantly**, σ^* (or ξ_k^* in $r > 2$ communities)–the quantities we are trying to output.

In several applications, the edge probabilities p and q (and a, b respectively) may be known by the algorithms, which makes the problem easier. When they are unknown, we need a reliable way to estimate them from the input.

Suppose that we have access to oracles that can provide us with these quantities. Let $\textsc{Oracle}_{\sigma^*}$ be the one that can provide us the true value of σ^* (or ξ_k^* when $r > 2$). Let $\textsc{Oracle}_{a,b}^{\alpha}$ be the one that can provide us the parameters $\hat{a}, \hat{b}$ accurately up to a factor of $1 \pm \alpha$ from the true values of a, b for a small constant $\alpha < 0.001$.

We present Algorithm 3 with the unrealistic assumption of the oracles. We then prove that Algorithm 3 ($\mathcal{M}_{\text{Stbl Oracle}}$) retains the privacy and utility of Algorithm 1. Because checking the $\mathcal{C}$-concentration can be done in polynomial time (in terms of n), and w.h.p. we do not invoke $d(G)$, the Algorithm takes polynomial time w.h.p.. After we confirm that $\mathcal{M}_{\text{Stbl Oracle}}$ has all the properties we need, we replace the oracles by realistic alternatives that we calculate from the input graph G. We then prove that Algorithm 2 that w.h.p. is the same as Algorithm 3 and inherits all of its properties.

Algorithm 2 $\mathcal{M}^f_{\text{Stbl Fast}}(G, \mathcal{C})$: Fast Stability Mechanism

1: $\hat{Y} \leftarrow f(G)$
2: $\sigma^* \leftarrow \hat{Y}$
3: $(\hat{a}, \hat{b}) \leftarrow$ *Algorithm 4*(G)
4: $\hat{\mathcal{C}} \leftarrow$ adjust $\mathcal{C}$ on $\alpha = 0.001$ to satisfy Proposition 4
5: Construct $\hat{\mathcal{C}}$-concentration using $\hat{C}, \sigma^*, \hat{a}, \hat{b}$
6: **if** G is $\hat{\mathcal{C}}$-concentrated **then**
7: $\quad \hat{d}(G) \leftarrow c\log n/\epsilon$
8: **else**
9: $\quad \hat{d}(G) \leftarrow \min(c\log n/\epsilon, d(G))$
10: **end if**
11: $\tilde{d}(G) \leftarrow \hat{d}(G) + \text{Lap}(1/\epsilon)$
12: **if** $\tilde{d}_f(G) > \frac{\log 1/\delta}{\epsilon}$ **then**
13: $\quad$ Output $\hat{Y}$
14: **else**
15: $\quad$ Output $\perp$
16: **end if**

Proposition 4. *In the context of Algorithm 3, if a graph G is $\hat{\mathcal{C}}$-concentrated then it is $\mathcal{C}$-concentrated.*

In each SBM setting, the proposition can be easily verified by checking all conditions of $\mathcal{C}$-concentration. $\textsc{Oracle}_{\sigma^*}$ guarantees us the true value of σ^*, so the differences between $\mathcal{C}$ and $\hat{\mathcal{C}}$ only come from the factor α of $\textsc{Oracle}_{a,b}^{\alpha}$. We use a tighter tuple of constants $\hat{\mathcal{C}}$ (compared to $\mathcal{C}$) to balance the fact that $\hat{a}$ and $\hat{b}$ may be off by some factor of $1 \pm \alpha$. This task can be done by adjusting each condition by scaling the respective c_k to a factor of $1 \pm 2\alpha$ in which direction makes the condition tighter.

Theorem 4. *(Complete proof in Lemma 27, 28, 31) $\mathcal{M}_{\text{Stbl Fast}}$ (Algorithm 2) is (ϵ, δ)-DP. When $\mathcal{M}^f_{\text{Stbl}}$ (Algorithm 1) achieves exact recovery (under some specific conditions of the SBMs under the view of Lemma 1), $\mathcal{M}^f_{\text{Stbl Fast}}$ also achieves exact recovery under the same conditions; and takes $O(poly(n))$ w.h.p..*

Proof sketch. $\mathcal{M}_{\text{Stbl Oracle}}$ differs from $\mathcal{M}_{\text{Stbl Fast}}$ by exact two steps: ($\sigma^* \leftarrow \textsc{Oracle}_{\sigma^*}$ line 2) and ($(\widehat{a}, \widehat{b}) \leftarrow \textsc{Oracle}_{a,b}^{\alpha}$ line 3). We then complete three big steps to prove the Theorem. **First,** we prove that $\Delta_{\hat{d}} = 1$ (the global sensitivity of $\hat{d}$) and follows the arguments of Theorem 5, substituting d by $\hat{d}$ to confirm the privacy guarantee of $\mathcal{M}_{\text{Stbl Oracle}}$. **Second,** we utilizes Proposition 4, arguing that $\hat{\mathcal{C}}$-concentration implies $\mathcal{C}$-concentration, to show that any graph G passes the test at line 6 is $\log n$-stable. Since $\hat{\mathcal{C}}$-concentration is a valid concentration, G passes the test at line 6 w.h.p.. In view of Lemma 1, $\mathcal{M}_{\text{Stbl Oracle}}$ achieve exact recovery. **Third,** we replace the oracles (by line 2, 3 of Algorithm 2), arguing that using $\hat{Y}$ is as good as $\textsc{Oracle}_{\sigma^*}$ w.h.p.. Similarly, the estimator by Algorithm 4 is at least as good as $\textsc{Oracle}_{\sigma^*}^{\alpha}$ w.h.p. in view of Lemma 30. We now can confirm that w.h.p., $\mathcal{M}_{\text{Stbl Fast}}$ is as good as $\mathcal{M}_{\text{Stbl Oracle}}$ and complete the proof.

6. Conclusion

Our work studied the community detection problem in SBMs under differential privacy, focusing on the theoretical boundaries for exact recovery. We show that, in three variants of SBMs: Binary Asymmetric, Censored Binary, and General Structure, exact recovery is possible by composing the Stability mechanism and the Semi-definite programming estimator. The main challenges lie in the design and analysis of $\mathcal{C}$-concentration–a key concept that determines the stability and optimality of the SDP Relaxation. Our results extend the best theoretical boundaries for exact recovery in SBMs for symmetric variants. We also propose the first polynomial time algorithms for SBMs under edge DP with arbitrary small ϵ that matches the non-private's recovery threshold when $\epsilon \to \infty$.

Acknowledgements. This research is supported by University of Virginia Strategic Investment Fund award number SIF160, NSF Grants OAC-1916805 (CINES), CCF-1918656 (Expeditions), IIS-1931628, IIS-1955797, CNS-2317193 and NIH grant R01GM109718.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Abbe, E., Bandeira, A. S., and Hall, G. Exact recovery in the stochastic block model. *IEEE Transactions on Information Theory*, 62(1):471–487, 2015.

Abbe, E., Fan, J., Wang, K., and Zhong, Y. Entrywise eigenvector analysis of random matrices with low expected rank. *Annals of statistics*, 48(3):1452, 2020.

Blocki, J., Blum, A., Datta, A., and Sheffet, O. Differentially private data analysis of social networks via restricted sensitivity. In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science*, ITCS '13, pp. 87–96, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450318594. doi: 10.1145/2422436.2422449. URL https://doi.org/10.1145/2422436.2422449.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008:10008, 2008.

Boppana, R. B. Eigenvalues and graph bisection: An average-case analysis. In *28th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 280–285, 1987.

Chen, H., Cohen-Addad, V., d'Orsi, T., Epasto, A., Imola, J., Steurer, D., and Tiegel, S. Private estimation algorithms for stochastic block models and mixture models. *arXiv preprint arXiv:2301.04822*, 2023.

Dwork, C., Roth, A., et al. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

Fortunato, S. Community detection in graphs. *Physics reports*, 486(3-5):75–174, 2010.

Gao, C., Ma, Z., Zhang, A. Y., and Zhou, H. H. Achieving optimal misclassification proportion in stochastic block models. *The Journal of Machine Learning Research*, 18 (1):1980–2024, 2017.

Girvan, M. and Newman, M. E. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

Guo, X., Li, X., Chang, X., and Ma, S. Privacy-preserving community detection for locally distributed multiple networks. *arXiv preprint arXiv:2306.15709*, 2023.

Hajek, B., Wu, Y., and Xu, J. Achieving exact cluster recovery threshold via semidefinite programming. *IEEE Transactions on Information Theory*, 62(5):2788–2797, 2016a.

Hajek, B., Wu, Y., and Xu, J. Achieving exact cluster recovery threshold via semidefinite programming: Extensions. *IEEE Transactions on Information Theory*, 62 (10):5918–5937, 2016b.

Hehir, J., Slavkovic, A., and Niu, X. Consistency of privacy-preserving spectral clustering under the stochastic block model. *arXiv preprint arXiv:2105.12615*, 2021.

Holland, P. W., Laskey, K. B., and Leinhardt, S. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

Imola, J., Murakami, T., and Chaudhuri, K. Locally differentially private analysis of graph statistics. In *30th USENIX Symposium on Security*, 2021.

Ji, T., Luo, C., Guo, Y., Ji, J., Liao, W., and Li, P. Differentially private community detection in attributed social networks. In *Asian Conference on Machine Learning*, pp. 16–31. PMLR, 2019.

Jiang, H., Pei, J., Yu, D., Yu, J., Gong, B., and Cheng, X. Applications of differential privacy in social network analysis: A survey. *IEEE transactions on knowledge and data engineering*, 35(1):108–127, 2021.

Karwa, V., Raskhodnikova, S., Smith, A., and Yaroslavtsev, G. Private analysis of graph structure. *Proceedings of the VLDB Endowment*, 4(11):1146–1157, 2011.

Kasiviswanathan, S. P., Nissim, K., Raskhodnikova, S., and Smith, A. Analyzing graphs with node differential privacy. In *Proceedings of the 10th Theory of Cryptography Conference on Theory of Cryptography*, TCC'13, pp. 457–476, Berlin, Heidelberg, 2013. Springer-Verlag. ISBN 978-3-642-36593-5. doi: 10.1007/978-3-642-36594-2_26. URL http://dx.doi.org/10.1007/978-3-642-36594-2_26.

Kiranmayi, M. and Maheswari, N. A review on privacy preservation of social networks using graphs. *Journal of Applied Security Research*, 16(2):190–223, 2021.

Lee, C. and Wilkinson, D. J. A review of stochastic block models and extensions for graph clustering. *Applied Network Science*, 4(1):1–50, 2019.

Li, Y., Purcell, M., Rakotoarivelo, T., Smith, D., Ranbaduge, T., and Ng, K. S. Private graph data release: A survey. *ACM Computing Surveys*, 55(11):1–39, 2023.

Massoulié, L. Community detection thresholds and the weak ramanujan property. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pp. 694–703, 2014.

McSherry, F. Spectral partitioning of random graphs. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 529–537, 2001.

Mohamed, M. S., Nguyen, D., Vullikanti, A., and Tandon, R. Differentially private community detection for stochastic block models. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 15858–15894. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/mohamed22a.html.

Mülle, Y., Clifton, C., and Böhm, K. Privacy-integrated graph clustering through differential privacy. In *EDBT/ICDT Workshops*, volume 157, 2015.

Nguyen, D. and Vullikanti, A. Differentially private densest subgraph detection. In *38th International Conference on Machine Learning (ICML)*. PMLR, July 2021.

Nguyen, D. and Vullikanti, A. Differentially private exact recovery for stochastic block models. *arXiv preprint arXiv:2406.02644*, 2024. URL https://arxiv.org/abs/2406.02644.

Nguyen, H. H., Imine, A., and Rusinowitch, M. Detecting communities under differential privacy. In *Proceedings of the 2016 ACM on Workshop on Privacy in the Electronic Society*, pp. 83–93, 2016.

Qin, T. and Rohe, K. Regularized spectral clustering under the degree-corrected stochastic blockmodel. *Advances in neural information processing systems*, 26, 2013.

Qin, Z., Yu, T., Yang, Y., Khalil, I., Xiao, X., and Ren, K. Generating synthetic decentralized social graphs with local differential privacy. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 425–438, 2017.

Seif, M., Goldsmith, A. J., and Poor, H. V. Differentially private community detection over stochastic block models with graph sketching. In *2023 57th Annual Conference on Information Sciences and Systems (CISS)*, pp. 1–6. IEEE, 2023.

Talagrand, M. Concentration of measure and isoperimetric inequalities in product spaces. *Publications Mathématiques de l'Institut des Hautes Etudes Scientifiques*, 81(1):73–205, 1995.

Tao, T. Topics in random matrix theory. *Graduate Studies in Mathematics*, 132, 2011.

Wang, P., Zhou, Z., and So, A. M.-C. A nearly-linear time algorithm for exact community recovery in stochastic block model. In *International Conference on Machine Learning (ICML)*, pp. 10126–10135. PMLR, 2020.

Zhou, B., Pei, J., and Luk, W. A brief survey on anonymization techniques for privacy preserving publishing of social network data. *ACM Sigkdd Explorations Newsletter*, 10(2):12–22, 2008.

A. Stability Mechanism

Theorem 5. *$\mathcal{M}^f_{\text{Stbl}}$ is (ϵ, δ)-differentially private*

Proof. (Adapted from Lemma 3.2 of (Mohamed et al., 2022) for completeness and consistency)

The proof that the stability based mechanism satisfies $(\epsilon, \delta))$- DP follows directly from (Dwork et al., 2014). Given a pair of neighbor graphs $G \sim G'$, $d(G)$ denotes the distance from G to its nearest unstable instance and $d(G')$ is the distance from G' to its nearest unstable instance. Due to the triangle inequality, $|d(G) - d(G')| \leq 1$, hence the sensitivity of d: $\Delta_d = 1$. Adding a Laplacian noise of magnitude of $1/\epsilon$ guarantees ϵ-differential privacy for $\tilde{d}$. In order to verify (ϵ, δ)-DP for the overall mechanism, we consider two scenarios: the first one, when the output of the mechanism is $\perp$. In this case, we have:

$$\Pr[\mathcal{M}^f_{\text{Stbl}}(G) = \perp] = \Pr\left[\tilde{d}(G) \leq \frac{\log 1/\delta}{\epsilon}\right] \tag{9}$$

$$\leq e^{\epsilon} \Pr\left[\tilde{d}(G') \leq \frac{\log 1/\delta}{\epsilon}\right] \tag{10}$$

$$= e^{\epsilon} \Pr[\mathcal{M}^f_{\text{Stbl}}(G') = \perp]. \tag{11}$$

where the first inequality follows from the fact that $\tilde{d}(G)$ satisfies ϵ-DP. For the second scenario, when the output of the mechanism $f(G)$, we have to analyze two cases.

The remaining part of the proof, we prove that output $f(G)$ in line 3 satisfies (ϵ, δ)-differential privacy to fulfill the proof of the theorem. We analyze two cases (1) $d(G) = 0$ and (2) $d(G) > 0$.

Case 1. $d(G) = 0$, we have $\Pr[\tilde{d}(G) > \frac{\log 1/\delta}{\epsilon}] = \Pr[Lap(1/\epsilon) > \frac{\log 1/\delta}{\epsilon}] \leq e^{-\log 1/\delta} = \delta$. For any set of output $S \subseteq (Range(f) \cup \{\perp\})$, we have

$$\begin{aligned}
\Pr[\mathcal{M}^f_{\text{Stbl}}(G) \in S] &\leq \Pr[\mathcal{M}^f_{\text{Stbl}}(G) \in (S \cup \{\perp\})] \\
&\leq \Pr[\mathcal{M}^f_{\text{Stbl}}(G) \in (S \cap \{\perp\})] + \Pr[\mathcal{M}^f_{\text{Stbl}}(G) \neq \perp] \\
&\leq \Pr[\mathcal{M}^f_{\text{Stbl}}(G) \in (S \cap \{\perp\})] + \delta \\
&\leq e^{\epsilon} \Pr[\mathcal{M}^f_{\text{Stbl}}(G') \in (S \cap \{\perp\})] + \delta \\
&\leq e^{\epsilon} \Pr[\mathcal{M}^f_{\text{Stbl}}(G') \in S] + \delta,
\end{aligned}$$

where the third inequality is because $\Pr[\mathcal{M}^f_{\text{Stbl}}(G) \neq \perp] = \Pr[\tilde{d}(G) > \frac{\log 1/\delta}{\epsilon}] \leq \delta$ and the fourth inequality is because $S \cap \{\perp\}$ is **(a)** $\emptyset$ or **(b)** $\{\perp\}$. When **(a)** happens, $\Pr[$
$M^f_{\text{Stbl}}(G) \in \emptyset] = \Pr[\mathcal{M}^f_{\text{Stbl}}(G') \in \emptyset] = 0$ and when **(b)** happens, it follows above proof that $\Pr[\mathcal{M}^f_{\text{Stbl}}(G) = \perp] \leq e^{\epsilon} \Pr[\mathcal{M}^f_{\text{Stbl}}(G') = \perp]$.

Case 2. $d(G) > 0$. In this case, G is at least 1-stable, which means: $\boldsymbol{\sigma}(G) = \boldsymbol{\sigma}(G') = \sigma$, we have:

$$\begin{aligned}
\Pr[\mathcal{M}^f_{\text{Stbl}}(G) = \sigma] &= \Pr[\tilde{d}(G) > \frac{\log 1/\delta}{\epsilon}] \\
&\leq e^{\epsilon} \Pr[\tilde{d}(G') > \frac{\log 1/\delta}{\epsilon}] \\
&= e^{\epsilon} \Pr[\mathcal{M}^f_{\text{Stbl}}(G') = \sigma],
\end{aligned}$$

and the Lemma follows. □

Lemma 6. *(Full version of Lemma 1) Given a function $f : \mathcal{G} \to \mathcal{R}$, the f-based Stability mechanism with $\delta = n^{-t}$ has $\Pr[M^f(G) \neq f(G)] \leq n^{-k_1} + n^{-k_2}$, if a graph G is $\frac{t+k_1}{\epsilon} \log n$-stable under function f with probability at least $1 - n^{-k_2}$. When $k_1, k_2 = \Omega(1)$, $\Pr[M^f(G) \neq f(G)] \leq n^{-\Omega(1)}$.*

Proof. From the definition of the Stability mechanism, $M^f(G)$ does not output $f(G)$ when $\tilde{d}(G) \leq \frac{\log 1/\delta}{\epsilon.}$. For simplicity, we denote $\tilde{d}(G)$ and $d(G)$ as $\tilde{d}, d$ respectively in the following equations. Hence, we have:

$$\Pr[M^f(G) \neq f(G)] = \Pr[\tilde{d} < \frac{\log 1/\delta}{\epsilon}] \tag{12}$$

$$= \Pr[d + \text{Lap}(1/\epsilon) < \frac{\log 1/\delta}{\epsilon}] \tag{13}$$

$$= \Pr[\text{Lap}(1/\epsilon) < \frac{\log 1/\delta}{\epsilon} - d] \tag{14}$$

$$= \Pr[\text{Lap}(1/\epsilon) < \frac{\log 1/\delta}{\epsilon} - d | d \geq \frac{t+k_1}{\epsilon}\log n] \Pr[d \geq \frac{t+k_1}{\epsilon}\log n] \tag{15}$$

$$+ \Pr[\text{Lap}(1/\epsilon) < \frac{\log 1/\delta}{\epsilon} - d | d < \frac{t+k_1}{\epsilon}\log n] \Pr[d < \frac{t+k_1}{\epsilon}\log n] \tag{16}$$

$$\leq \Pr[\text{Lap}(1/\epsilon) < \frac{\log 1/\delta}{\epsilon} - d | d \geq \frac{t+k_1}{\epsilon}\log n] + \Pr[d < \frac{t+k_1}{\epsilon}\log n] \tag{17}$$

$$\leq \Pr[\text{Lap}(1/\epsilon) < \frac{\log 1/\delta}{\epsilon} - d | d \geq \frac{t+k_1}{\epsilon}\log n] + n^{-k_2} \tag{18}$$

$$\leq \Pr[\text{Lap}(1/\epsilon) < \frac{t\log n - (t+k_1)\log n}{\epsilon}] + n^{-k_2} \tag{19}$$

$$\leq \Pr[\text{Lap}(1/\epsilon) < \frac{-k_1 \log n}{\epsilon}] + n^{-k_2} \tag{20}$$

$$= \Pr[\text{Lap}(1/\epsilon) < \frac{-k_1 \log n}{\epsilon}] + n^{-k_2} \tag{21}$$

$$\leq n^{-k_2} + n^{-k_2}, . \tag{22}$$

and the Lemma follows. □

B. Binary Asymmetric SBM (BASBM)

In this section we examine the Binary Asymmetric SBM (referred to as BASBM) where the sizes of the clusters are defined by a parameter $\rho \in (0, 1/2)$. We assume that the first cluster has size $K = \rho n$ and the second cluster has size $n - K = (1-\rho)n$.

Problem Formulation and Definitions We adapt (Hajek et al., 2016b)'s analyses to formulate the Semi-definite program to solve the community detection in the BASBM.

Definition 4. *We define quantities in our analysis as follows:*

- $Y = \sigma\sigma^T$
- $Y^* = \sigma^*\sigma^{*T}$
- $\tau = \frac{a-b}{\log a - \log b}$
- $\lambda^* = \tau \log n/n$
- $d^* \in \mathbb{R}^n : d_i^* = \sum_{j=1}^n A_{ij}\sigma_i^*\sigma_j^* - \lambda^*(2K-n)\sigma_i^*$
- $D^* = diag\{d^*\}$
- I *is the identity matrix*
- J *is the all-one matrix*
- $\mathcal{J}(x) = \left(\lambda^* - \frac{p+q}{2}\right) x^\top Jx$
- $\check{x} = \arg\max_x x^\top Jx$ *subject to* $\|x\|_2 = 1$ *and* $\langle x, \sigma^* \rangle = 0$

- $h(\alpha) = a\rho + b(1-\rho) - \sqrt{\alpha^2 + 4\rho(1-\rho)ab} + \frac{|\alpha|}{2}\log\frac{\rho b}{(1-\rho)a}$
- $\tilde{h}(x) = a\rho + b(1-\rho) - \sqrt{(\tau(1-2\rho)-x)^2 + 4\rho(1-\rho)ab} + \frac{\tau(1-2\rho)-x}{2}\log\frac{\rho b}{(1-\rho)a}$

Definition 5. ***Definition of*** **Concentration.**

We assume a graph G is generated by a Binary Symmetric SBM with two communities sized ρn and $(1-\rho)n$ for some constant ρ. The link between two endpoints from the same community are generated with probability $p = \frac{a\log n}{n}$ and the link between two endpoints from different communities are generated with probability $q = \frac{b\log n}{n}$.

G is called (c_1, c_2, c_3, c_4)-concentrated for some constants $c_i > 0$ if G satisfies all 4 conditions:

- $\|A - \mathbb{E}[A]\|_2 \leq c_1\sqrt{\log n}$
- $\check{x}^\top D^* \check{x} + \mathcal{J}(\check{x}) > c_2 \log n$
- $\|(D^* - \mathbb{E}[D^*])\check{x}\|_2 \leq c_3\sqrt{\log n}$
- $d_i^* \geq c_4 \log n$ *for every* $i \in [n]$

Assumptions of the parameters

We have several assumptions of the parameters of the SBM. These assumptions will be used in our analyses to derive the conditions in which Exact Recovery under Differential Privacy is feasible.

The exists a constant $c > 0$ such that

$$\tilde{h}(c) > 1 \text{ or equivalently, } h(c - \tau(1-2\rho)) > 1 \tag{23}$$

In other words, we can say that:

$$\tilde{h}(c) = 1 + \Omega(1) \text{ or equivalently, } h(c - \tau(1-2\rho)) = 1 + \Omega(1) \tag{24}$$

The Binary Asymmetric SBM can be solve in the non-privacy setting by solving the following SDP relaxation. We denote $SDP(G)$ as a function taking input graph G and outputting the optimal solution of the SDP relaxation constructed by its adjacency matrix $A(G)$.

Definition 6. *SDP Relaxation of the Binary Asymmetric SBM:*

$$\widehat{Y}_{SDP} = \arg\max_Y \langle A, Y\rangle \tag{25}$$
$$\text{s.t. } Y \succcurlyeq 0 \tag{26}$$
$$Y_{ii} = 1, \text{ for } i \in [n] \tag{27}$$
$$\langle J, Y\rangle = (2K - n)^2 \tag{28}$$

Lemma 7. ***Theorem 5 of (Hajek et al., 2016a).*** *Let A be a symmetric and zero-diagonal random matrix, where the entries $A_{ij} (i < j)$ are independent and $[0,1]$-valued. Assume that $\mathbb{E}[A_{ij}] \leq p$, where $c_0 \log n/n \leq p \leq 1 - c_1$ for arbitrary constants $c_0, c_1 > 0$. Then for any $c > 0$, there exists $c' > 0$ such that for any $n \geq 1, \Pr[\|A - \mathbb{E}[A]\|_2 \leq c'\sqrt{np}] \geq 1 - n^{-c}$.*

Lemma 8. ***Theorem 2.1.13 of (Tao, 2011)*** *(Originally from (Talagrand, 1995)). Let $P > 0$, and let $X_1, \ldots, X_n$ be independent complex variables with $|X_i| \leq P$ for all $1 \leq i \leq n$. Let $F : \mathbf{C}^n \to \mathbf{R}$ be a 1-Lipschitz convex function. Then for any λ one has*

$$\Pr[|F(X) - \mathbb{E}[F(X)]| \geq \lambda P] \leq C\exp(-c\lambda^2) \tag{29}$$

for some absolute constants $C, c > 0$.

Lemma 9. ***Lemma 2 of (Hajek et al., 2016b).*** *Suppose $a, b > 0, \alpha \in \mathbb{R}$, and $\rho_1, \rho_2 > 0$. Let X, R be independent with $X \sim Binom(m_1, \frac{a \log n}{n})$ and $R \sim Binom(m_2, b)$, where $m_1 = \rho_1 n + o(n)$ and $m_2 = \rho_2 n + o(n)$ as $n \to \infty$. Let $k \in \mathbb{I}$ such that $k = \alpha \log n + o(\log n)$. If $\alpha \leq a\rho_1 - b\rho_2$,*

$$\Pr[X - R \leq k] = n^{-g(\rho_1, \rho_2, a, b, \alpha) + o(1)}, \tag{30}$$

where $g(\rho_1, \rho_2, a, b, alpha) = a\rho_1 + b\rho_2 - \gamma - \frac{\alpha}{2} \log \frac{(\gamma - \alpha) a \rho_1}{(\gamma + \alpha) b \rho_2}$ with $\gamma = \sqrt{\alpha^2 + 4\rho_1 \rho_2 ab}$.

Furthermore, for any $m_1, m_2, \in \mathbb{N}$, $k \in \mathbb{I}$ such that $k < (m_1 a - m_2 b) \log n / n$,

$$\Pr[X - R \leq k] \leq n^{-g(m_1/n, m_2/n, a, b, k/\log n)} \tag{31}$$

Lemma 10. *A graph G generated by a Binary Asymmetric SBM with two communities sized ρn and $(1-\rho)n$ for some constant ρ and with $p = \frac{a \log n}{n}$ and $q = \frac{b \log n}{n}$, there exists some constant $c_2 : 0 < c_2 < \tau - b$ such that*

$$\check{x}^\top D^* \check{x} + \mathcal{J}(\check{x}) > c_2 \log n, \tag{32}$$

with probability at least $1 - n^{-\Omega(1)}$.

Proof. **The second condition** relies on Talagrand's concentration inequality for Lipschitz convex functions (Lemma 8). For the context of Lemma 8, set the function $F(A) = \check{x}^\top D^* \check{x}$ where the adjacency matrix A is the argument of the function F (as D^* can be represented as a mapping of A). The next step is to prove that $F(A)$ is Lipschitz continuous in A.

We note that $\check{x}^\top D^* \check{x} = \langle A, B \rangle - \lambda^* (2K - n) \sum_{i=1}^n \check{x}_i^2 \sigma_i^*$ where $B_{ij} = \sigma_i^* \sigma_j^* \check{x}_i^2$, which makes $\check{x}^\top D^* \check{x}$ is Lipchitz continuous with Lipschitz constant $\|B\|_F = \sqrt{\frac{(1-\rho)^2}{\rho} + \frac{\rho^2}{1-\rho}} + o(1)$We treat all A_{ij} as X in Lemma 8 with $P = 1$ (as $0 \leq A_{ij} \leq 1$). By Lemma 8, for any c there exists some constants C, c_2' and $\lambda = c_2' \sqrt{\log n}$ that:

$$\Pr[|\check{x}^\top D^* \check{x} - \mathbb{E}[\check{x}^\top D^* \check{x}]| \geq c_2' \sqrt{\log n}] \leq C \exp(-c \log n) \tag{33}$$

It follows that if we pick any constant $c > 0$, then with probability at least $1 - n^{-\Omega(1)}$, we have:

$$\check{x}^\top D^* \check{x} - \mathbb{E}[\check{x}^\top D^* \check{x}] > -c_2' \sqrt{\log n} \tag{34}$$

We now analyze $\mathcal{J}(\check{x})$. Because $\check{x}^\top J \check{x} = 4K(n-K)/n$, we have:

$$\begin{aligned}
\mathcal{J}(\check{x}) &= \left(\lambda^* - \frac{p+q}{2}\right) \check{x}^T J \check{x} && (35)\\
&= \left(\tau - \frac{a+b}{2}\right) 4K(n-K) \log n / n^2 && (36)\\
&= (2\tau - a - b) 2K(n-K) \log n / n^2 && (37)\\
&= (\tau - a) 2K(n-K) \log n / n^2 + (\tau - b) 2K(n-K) \log n / n^2 && (38)
\end{aligned}$$

And then we analyze $\mathbb{E}[\check{x}^\top D^* \check{x}]$:

In the first case: $\sigma_i^* = 1$:

$$\mathbb{E}[d_i^*] \stackrel{\text{def}}{=} \bar{d}_+ = (K(a - \tau) + (n - K)(\tau - b) - a) \log n / n \tag{39}$$

In the second case: $\sigma_i^* = -1$:

$$\mathbb{E}[d_i^*] \stackrel{\text{def}}{=} \bar{d}_- = ((n-K)(a-\tau) + (K(\tau-b)-a)\log n/n \tag{40}$$

Then we have:

$$\begin{aligned}
\mathbb{E}[\check{x}^\top D^* \check{x}] &= (n-K)\bar{d}_+/n + K\bar{d}_-/n && (41)\\
&= \big(2K(n-K)(a-\tau) + (K^2 + (n-K)^2)(\tau-b) - na\big)\log n/n^2 && (42)\\
&= 2K(n-K)(a-\tau)\log n/n^2 + K^2(\tau-b)\log n/n^2 + (n-K)^2(\tau-b)\log n/n^2 - a\log n/n && (43)
\end{aligned}$$

.

Compose with the result of equation 38, taking the sum of the two quantities, we have:

$$\begin{aligned}
\mathbb{E}[\check{x}^\top D^* \check{x}] + \mathcal{J}(\check{x}) &= \frac{(\tau-b)\log n}{n^2}\left(K^2 + (n-K)^2 + 2K(n-K)\right) - \frac{a\log n}{n} && (44)\\
&= (\tau-b)\log n - \frac{a\log n}{n} && (45)
\end{aligned}$$

Back to the second condition, composing the results of equations 34, 43, 38, 45, we rewrite the second condition as follows:

$$\begin{aligned}
\check{x}^\top D^* \check{x} + \mathcal{J}(\check{x}) &= \check{x}^\top D^* \check{x} - \mathbb{E}[\check{x}^\top D^* \check{x}] + E[\check{x}^\top D^* \check{x}] + \mathcal{J}(\check{x}) && (46)\\
&\geq -c_2'\sqrt{\log n} + (\tau-b)\log n - \frac{a\log n}{n} && (47)\\
&> c_2 \log n, && (48)
\end{aligned}$$

for some constants $0 < c_2 < \tau - b$ and n large enough. Since equation 34 holds with probability at least $1 - n^{-\Omega(1)}$, the Lemma follows. □

Lemma 11. *A graph G generated by a Binary Asymmetric SBM with two communities sized ρn and $(1-\rho)n$ for some constant ρ and with $p = \frac{a\log n}{n}$ and $q = \frac{b\log n}{n}$, there exists some tuples of constant c_3 such that*

$$\|(D^* - \mathbb{E}[D^*]\|_2)\check{x} \leq c_3\sqrt{\log n}, \tag{49}$$

with probability at least $1 - n^{-\Omega(1)}$.

Proof. **The third condition** can be constructed in a similar way to the second condition. First, we have:

$$\begin{aligned}
\|(D^* - \mathbb{E}[D^*])\check{x}\|_2^2 &= \sum_i \left(\sum_j (A_{ij} - \mathbb{E}[A_{ij}])\sigma_i^*\sigma_j^*\check{x} - i\right)^2 && (50)\\
&= \sum_i \check{x}_i^2 \sigma_i^{*2}\left(\sum_j (A_{ij} - \mathbb{E}[A_{ij}])\sigma_j^*\right)^2 && (51)\\
&= \sum_i \check{x}_i^2 \left(\sum_j (A_{ij} - \mathbb{E}[A_{ij}])\sigma_j^*\right)^2 && (52)
\end{aligned}$$

Then we analyze the expectation of $\|(D^* - \mathbb{E}[D^*])\check{x}\|_2$:

$$\mathbb{E}[\|(D^* - \mathbb{E}[D^*])\check{x}\|_2] \leq \sqrt{\mathbb{E}\left[\|(D^* - \mathbb{E}[D^*])\check{x}\|_2^2\right]} \tag{53}$$

$$= \sqrt{\mathbb{E}\left[\sum_i \check{x}_i^2 \left(\sum_j (A_{ij} - \mathbb{E}[A_{ij}])\sigma_j^*\right)^2\right]} \tag{54}$$

$$= \sqrt{\sum_i \check{x}_i^2 \mathbb{E}\left[\left(\sum_j (A_{ij} - \mathbb{E}[A_{ij}])\sigma_j^*\right)^2\right]} \tag{55}$$

$$\overset{(a)}{=} \sqrt{\sum_i \check{x}_i^2 \sum_j \text{Var}(A_{ij})} \tag{56}$$

$$\leq \sqrt{\sum_i \check{x}_i^2 \sum_j \frac{a\log n}{n}\left(1 - \frac{a\log n}{n}\right)} \tag{57}$$

$$\leq \sqrt{\sum_i \check{x}_i^2 a\log n} \tag{58}$$

$$= \sqrt{a\log n \sum_i \check{x}_i^2} \tag{59}$$

$$= \sqrt{a\log n}, \tag{60}$$

where (a) is because of:

$$\mathbb{E}\left[\left(\sum_j (A_{ij} - \mathbb{E}[A_{ij}])\sigma_j^*\right)^2\right] \tag{61}$$

$$= \left(\mathbb{E}\left[\sum_j (A_{ij} - \mathbb{E}[A_{ij}])\sigma_j^*\right]\right)^2 + \text{Var}\left(\sum_j (A_{ij} - \mathbb{E}[A_{ij}])\sigma_j^*\right) \tag{62}$$

$$\overset{(b)}{=} \left(\sum_j \mathbb{E}[(A_{ij} - \mathbb{E}[A_{ij}])]\sigma_j^*\right)^2 + \sum_j \text{Var}\left((A_{ij} - \mathbb{E}[A_{ij}])\sigma_j^*\right) \tag{63}$$

$$= 0 + \sum_j \sigma_j^{*2}\text{Var}(A_{ij} - \mathbb{E}[A_{ij}]) \tag{64}$$

$$= \sum_j \text{Var}(A_{ij}), \tag{65}$$

where (b) is because of linearity of expectations and A_{ij}s are independent.

Next we prove that $\|(D^* - \mathbb{E}[D^*])\check{x}\|_2$ is convex and Lipschitz continuous in A with Lipschitz constant bounded by $\max\left\{\sqrt{\frac{1-\rho}{\rho}}, \sqrt{\frac{\rho}{1-\rho}}\right\}$. For any A, A', let $D^*, D^{*\prime}$ be the respectively diagonal matrices, we have:

$$\left|\|(D^* - \mathbb{E}[D^*]\check{x})\|_2 - \|(D^{*\prime} - \mathbb{E}[D^{*\prime}]\check{x})\|_2\right| \leq \|(D^* - D^{*\prime})\check{x}\|_2 \tag{66}$$

$$= \sqrt{\sum_i \check{x}_i^2 \left(\sum_j (A_{ij} - A'_{ij})\sigma_j^*\right)^2} \tag{67}$$

$$\leq \sqrt{\sum_i \left(\sum_j (A_{ij} - A'_{ij})\sigma_j^*\right)^2} \max\left\{\sqrt{\frac{n-K}{nK}}, \sqrt{\frac{K}{n(n-K)}}\right\} \tag{68}$$

$$\leq \|A - A'\|_F \max\left\{\sqrt{\frac{1-\rho}{\rho}}, \sqrt{\frac{\rho}{1-\rho}}\right\} \tag{69}$$

Then we apply Lemma 8, with $P = 1$, for any $c > 0$ there exists $C, c'_3 > 0$ and $\lambda = c'_3\sqrt{\log n}$ such that:

$$\Pr[|\|(D^* - \mathbb{E}[D^*])\check{x}\|_2 - \mathbb{E}[\|(D^* - \mathbb{E}[D^*])\check{x}\|_2]| \geq c'_3\sqrt{\log n}] \leq C\exp(c\log n) \tag{70}$$

Choosing a constant $c > 0$, with probability at least $1 - n^{-\Omega(1)}$, we have:

$$\|(D^* - \mathbb{E}[D^*])\check{x}\|_2 - \mathbb{E}[\|(D^* - \mathbb{E}[D^*])\check{x}\|_2 < c'_3\sqrt{\log n} \tag{71}$$

As before we prove that $\mathbb{E}[\|(D^* - \mathbb{E}[D^*])\check{x}\|_2 \leq \sqrt{a\log n}$, setting $c_3 = c'_3 + \sqrt{a}$, we have that with probability at least $1 - n^{-\Omega(1)}$:

$$\|(D^* - \mathbb{E}[D^*])\check{x}\|_2 \leq c_3\sqrt{\log n} \tag{72}$$

□

Lemma 12. *Let function h and g be defined as above, with fixed values of τ, ρ, a, b and some constants c_4, we have:*

$$h(-\tau(1-2\rho) + c4) < g(\rho, 1-\rho, a, b, -\tau(1-2\rho) + c_4) \text{ ,and} \tag{73}$$

$$h(\tau(1-2\rho) + c4) < g(1-\rho, \rho, a, b, -\tau(1-2\rho) + c_4). \tag{74}$$

Furthermore, for function $h(\alpha)$, if $|\alpha_1| < |\alpha_2|$ then $h(\alpha_1) > h(\alpha_2)$.

Proof. We prove the first inequality as follows: setting $\gamma = \sqrt{(-\tau(1-2\rho) + c_4)^2 + 4\rho(1-\rho)ab}$ the definition of g, if $c4 < \tau(1-2\rho)$ we have:

$$g(\rho, 1-\rho, a, b, -\tau(1-2/\rho)+c_4) \tag{75}$$
$$=a\rho + b(1-\rho) - \gamma - \frac{-\tau(1-2\rho)+c_4}{2}\log\frac{(\gamma-(-\tau(1-2\rho)+c_4))a\rho}{(\gamma+(-\tau(1-2\rho)+c_4))b(1-\rho)} \tag{76}$$
$$=a\rho + b(1-\rho) - \gamma + \frac{\tau(1-2\rho)-c_4}{2}\log\frac{(\gamma+(\tau(1-2\rho)-c_4))a\rho}{(\gamma-(\tau(1-2\rho)-c_4))b(1-\rho)} \tag{77}$$
$$\geq a\rho + b(1-\rho) - \gamma + \frac{\tau(1-2\rho)-c_4}{2}\log\frac{a\rho}{b(1-\rho)},\ \text{since } \tau(1-2\rho)-c_4 \geq 0 \tag{78}$$
$$\geq a\rho + b(1-\rho) - \gamma + \frac{\tau(1-2\rho)-c_4}{2}\log\frac{b\rho}{a(1-\rho)},\ \text{since } a > b \tag{79}$$
$$=a\rho + b(1-\rho) - \sqrt{(\tau(1-2\rho)-c_4)^2 + 4\rho(1-\rho)ab} + \frac{\tau(1-2\rho)-c_4}{2}\log\frac{b\rho}{a(1-\rho)} \tag{80}$$
$$=h(-\tau(1-2\rho)+c_4). \tag{81}$$

In case $c_4 > \tau(1-2\rho)$, we have:

$$g(\rho, 1-\rho, a, b, -\tau(1-2/\rho)+c_4) \tag{82}$$
$$=a\rho + b(1-\rho) - \gamma - \frac{-\tau(1-2\rho)+c_4}{2}\log\frac{(\gamma-(-\tau(1-2\rho)+c_4))a\rho}{(\gamma+(-\tau(1-2\rho)+c_4))b(1-\rho)} \tag{83}$$
$$=a\rho + b(1-\rho) - \gamma + \frac{\tau(1-2\rho)-c_4}{2}\log\frac{(\gamma+(\tau(1-2\rho)-c_4))a\rho}{(\gamma-(\tau(1-2\rho)-c_4))b(1-\rho)} \tag{84}$$
$$\geq a\rho + b(1-\rho) - \gamma + \frac{\tau(1-2\rho)-c_4}{2}\log\frac{a\rho}{b(1-\rho)},\ \text{since } \tau(1-2\rho)-c_4 \leq 0 \tag{85}$$
$$\geq a\rho + b(1-\rho) - \gamma + \frac{\tau(1-2\rho)-c_4}{2}\log\frac{a(1-\rho)}{b\rho},\ \text{since } 1-\rho > \rho \tag{86}$$
$$=a\rho + b(1-\rho) - \gamma + \frac{c_4-\tau(1-2\rho)}{2}\log\frac{b\rho}{a(1-\rho)} \tag{87}$$
$$=a\rho + b(1-\rho) - \sqrt{(\tau(1-2\rho)-c_4)^2 + 4\rho(1-\rho)ab} + \frac{c_4-\tau(1-2\rho)}{2}\log\frac{b\rho}{a(1-\rho)} \tag{88}$$
$$=h(-\tau(1-2\rho)+c_4). \tag{89}$$

Similarly, the second inequality can be proved as follows: setting $\gamma = \sqrt{(\tau(1-2\rho)+c_4)^2 + 4\rho(1-\rho)ab}$ (note that this γ is different from the one in the first inequality above), we have:

$$g(1-\rho, \rho, a, b, \tau(1-2/\rho)+c_4) \tag{90}$$
$$=a(1-\rho) + b\rho - \gamma - \frac{\tau(1-2\rho)+c_4}{2}\log\frac{(\gamma-(\tau(1-2\rho)+c_4))a(1-\rho)}{(\gamma+(\tau(1-2\rho)+c_4))b\rho} \tag{91}$$
$$=a(1-\rho) + b\rho - \gamma + \frac{\tau(1-2\rho)+c_4}{2}\log\frac{(\gamma+(\tau(1-2\rho)+c_4))b\rho}{(\gamma-(\tau(1-2\rho)+c_4))a(1-\rho)} \tag{92}$$
$$\geq a(1-\rho) + b\rho - \gamma + \frac{\tau(1-2\rho)+c_4}{2}\log\frac{b\rho}{a(1-\rho)},\ \text{since } \tau(1-2\rho)+c_4 \geq 0 \tag{93}$$
$$\geq a\rho + b(1-\rho) - \gamma + \frac{\tau(1-2\rho)+c_4}{2}\log\frac{b\rho}{a(1-\rho)},\ \text{since } a > b \text{ and } \rho \leq 1-\rho \tag{94}$$
$$=h(\tau(1-2\rho)+c_4). \tag{95}$$

Finally, for $|\alpha_1| < |\alpha_2|$, we have:

$$h(\alpha_1) - h(\alpha_2) = -\sqrt{\alpha_1^2 + 4\rho(1-\rho)ab} + \frac{|\alpha_1|}{2}\log\frac{b\rho}{a(1-\rho)} + \sqrt{\alpha_2^2 + 4\rho(1-\rho)ab} - \frac{|\alpha_2|}{2}\log\frac{b\rho}{a(1-\rho)} \tag{96}$$

$$> 0 + \frac{\log\frac{b\rho}{a(1-\rho)}}{2}(|\alpha_1| - |\alpha_2|) \tag{97}$$

$$\geq 0, \tag{98}$$

where the last inequality is because $\frac{b\rho}{a(1-\rho)} < 1$ then $\log\frac{b\rho}{a(1-\rho)} < 0$ and $|\alpha_1| - |\alpha_2| < 0$.

□

Lemma 13. *A graph G generated by a Binary Asymmetric SBM with two communities sized ρn and $(1-\rho)n$ for some constant ρ and with $p = \frac{a\log n}{n}$ and $q = \frac{b\log n}{n}$, there exists some tuples of constant c_4 such that*

$$\min_{i\in[n]} d_i^* \geq c_4 \log n, \tag{99}$$

with probability at least $1 - n^{-\Omega(1)}$.

Proof. **The fourth condition** will be analyzed in two cases: for nodes in the first cluster ($\sigma_i^* = 1$) and for nodes in the second cluster ($\sigma_i^* = -i$).

The first case: $\sigma_i^* = 1$. We first set two random variables $X \sim Binom(K-1, \frac{a\log n}{n})$ and $R \sim Binom(n-K, \frac{b\log n}{n})$. Setting $m_1 = K-1$, $m_2 = n-K$, $k = -\tau(1-2\rho)\log n + c_4\log n, \alpha = -(\tau(1-2\rho)+c_4)$. Fix a node i in the first cluster, we have $\sum_j A_{ij}\sigma_i^*\sigma_j^* = X - R$. Applying Lemma 9 and selecting some constants $c_4 > 0$) that satisfies Equation 24, we have:

$$\Pr\left[\sum_j A_{ij}\sigma_i^*\sigma_j^* \leq -\tau(1-2\rho)\log n + c_4\log n\right] \leq n^{-g(\rho,1-\rho,a,b,-\tau(1-2\rho)+c_4)} \tag{100}$$

$$\overset{(a)}{\leq} n^{-h(-\tau(1-2\rho)+c_4)} \tag{101}$$

$$\overset{(b)}{=} n^{-\tilde{h}(c_4)} \tag{102}$$

$$\overset{(c)}{\leq} n^{-1-\Omega(1)}, \tag{103}$$

where (a) is because of the result Lemma 12 such that $h(-\tau(1-2\rho)+c_4) \leq g(\rho, 1-\rho, a, b, -\tau(1-2\rho)+c_4)$, and (b) is because of the definitions of h and $\tilde{h}$, and (c) is the assumption at equation (24).

With the assumption that $\tilde{h}(c_4) > 1$, applying union bound on all nodes i in the first cluster we have that with probability at least $1 - n^{-\Omega(1)}$:

$$\sum_j A_{ij}\sigma_i^*\sigma_j^* > -\tau(1-2\rho)\log n + c_4\log n \tag{104}$$

Recall that $d_i^* = \sum_j A_{ij}\sigma_i^*\sigma_j^* - \sigma_i^*\tau(2K-n)\log n/n$, with $\sigma_i^* = 1$, we have:

$$d_i^* = \sum_j A_{ij}\sigma_i^*\sigma_j^* + \tau(1-2\rho)\log n \tag{105}$$

$$> -\tau(1-2\rho)\log n + c_4\log n + \tau(1-2\rho)\log n \tag{106}$$

$$= c_4\log n. \tag{107}$$

The second case: $\sigma_i^* = -1$. We set two random variables $X \sim Binom(n-K-1, \frac{a\log n}{n})$ and $R \sim Binom(K, \frac{b\log n}{n})$. Setting $m_1 = n-K-1$, $m_2 = K$, $k = \tau(1-2\rho)\log n + c_4 \log n$, $\alpha = \tau(1-2\rho)+c_4$. Fix a node i in the second cluster, we have $\sum_j A_{ij}\sigma_i^*\sigma_j^* = X - R$. Applying Lemma 9, we have:

$$\Pr\left[\sum_j A_{ij}\sigma_i^*\sigma_j^* \le \tau(1-2\rho)\log n + c_4\log n\right] \le n^{-g(1-\rho,\rho,a,b,\tau(1-2\rho)+c_4)} \tag{108}$$

$$\overset{(a)}{\le} n^{-h(\tau(1-2\rho)+c_4)} \tag{109}$$

$$\overset{(b)}{\le} n^{-h(-\tau(1-2\rho)+c_4)} \tag{110}$$

$$\overset{(c)}{=} n^{-\tilde{h}(c_4)} \tag{111}$$

$$\overset{(d)}{\le} n^{-1-\Omega(1)}, \tag{112}$$

where (a) is because of the result Lemma 12 such that $h(\tau(1-2\rho)+c_4) \le g(1-\rho,\rho,a,b,\tau(1-2\rho)+c_4)$, and (b) is because of the result of Lemma 12 such that such that $h(-\tau(1-2\rho)+c_4) \le h(\tau(1-2\rho)+c_4)$, and (c) is because of the definitions of h and $\tilde{h}$, and (d) is the assumption at equation (24).

With the assumption that $\tilde{h}(c_4) > 1$, applying union bound on all nodes i in the second cluster we have that with probability at least $1 - n^{-\Omega(1)}$:

$$\sum_j A_{ij}\sigma_i^*\sigma_j^* > \tau(1-2\rho)\log n + c_4\log n \tag{113}$$

Recall that $d_i^* = \sum_j A_{ij}\sigma_i^*\sigma_j^* - \sigma_i^*\tau(2K-n)\log n/n$, with $\sigma_i^* = -1$, we have:

$$d_i^* = \sum_j A_{ij}\sigma_i^*\sigma_j^* - \tau(1-2\rho)\log n \tag{114}$$

$$> \tau(1-2\rho)\log n + c_4\log n - \tau(1-2\rho)\log n \tag{115}$$

$$= c_4\log n. \tag{116}$$

Composing equations 107 and 116 we have that with probability at least $1 - n^{-\Omega(1)}$:

$$\min_{i\in[n]} d_i^* > c_4\log n \tag{117}$$

□

Lemma 14. *A graph G generated by a Binary Asymmetric SBM with two communities sized ρn and $(1-\rho)n$ for some constant ρ and with $p = \frac{a\log n}{n}$ and $q = \frac{b\log n}{n}$, there exists some tuples of constants c_1, c_2, c_3, c_4 such that G is (c_1, c_2, c_3, c_4)-concentrated with probability at least $1 - n^{-\Omega(1)}$.*

Proof. In this analysis, we inherit some analyses from Lemma 3 and Theorem 1 of (Hajek et al., 2016b). We use different bounds on the Second and the Fourth conditions, which are stronger than similar bounds in (Hajek et al., 2016b).

The first condition can be derived directly from (Hajek et al., 2016a)'s Theorem 5. We adapt ((Hajek et al., 2016a), Theorem 5) as Lemma 7 for the convenience of our analyses. Since each edge of G is generated with probability at least $b\log n/n$ and at most $a\log n/n$, by Lemma 7, with probability at least $1 - n^{-\Omega(1)}$, there exists some constant c_1 such that:

$$\|A - \mathbb{E}[A]\|_2 \leq c_1\sqrt{n\frac{\log n}{n}} = c_1\sqrt{\log n} \tag{118}$$

Applying the union bound on the statement above and the results of Lemma 10 (for **Second condition**), Lemma 11 (**Third condition**), and Lemma 13 (**Fourth condition**), there exists some tuples of constants c_i such that G is (c_1, c_2, c_3, c_4)-concentrated with probability at least $1 - n^{-\Omega(1)}$.

□

Lemma 15. *If G is (c_1, c_2, c_3, c_4)-concentrated then for every graph $G' : d(G, G') < \frac{c}{\epsilon}\log n$, i.e., G' can be constructed by flipping at most $\frac{c}{\epsilon}\log n$ connections of G, G' is (c_1', c_2', c_3', c_4')-concentrated, where $c_1' = c_1 + \sqrt{2c/\epsilon}, c_2' = c_2 - c/\epsilon, c_3' = c_3 + \sqrt{2c(1-\rho)/\epsilon\rho}, c_4' = c_4 - c/\epsilon$.*

Proof. Given the graph G is (c_1, c_2, c_3, c_4)-concentrated, it means that the graph G satisfies all four conditions of the *concentration* notation. We prove that all graph G' that can be constructed by flipping up to $\frac{c\log n}{\epsilon}$ connections of G will also satisfy all four conditions of *concentration*, albeit with slightly different tuples of constants.

The first condition: We follow (Mohamed et al., 2022) to prove that G' will satisfy the first condition. Let $\bar{A}$ be the expected adjacency matrix of graphs generated by the SBM. For both G and G' generated by the same SBM, we have $\mathbb{E}[A] = \mathbb{E}[A'] = \bar{A}$. Also, because G' can be formed by flipping up to $\frac{c\log n}{\epsilon}$ connections of G, then $\|A - A'\|_F \leq \sqrt{2c\log n/\epsilon}$. Now consider the ℓ_2-norm of the difference between A' and $\mathbb{E}[A']$:

$$\begin{aligned}
\|A' - \mathbb{E}[A']\|_2 &= \|A' - \bar{A}\|_2 && (119)\\
&= \|A' - A + A - \bar{A}\|_2 && (120)\\
&\leq \|A - \bar{A}\| + \|A' - A\|_2 && (121)\\
&\leq c_1\sqrt{\log n} + \|A' - A\|_F && (122)\\
&\leq c_1\sqrt{\log n} + \sqrt{2c\log n/\epsilon} && (123)\\
&= (c_1 + \sqrt{2c/\epsilon})\sqrt{\log n} && (124)\\
&= c_1'\sqrt{\log n}. && (125)
\end{aligned}$$

The second condition: We revisit the second condition: $\check{x}^\top D^*\check{x} + \mathcal{J}(\check{x}) > c_2\log n$ holds for the graph G, where $\mathcal{J}(\check{x}) = \left(\lambda^* - \frac{p+q}{2}\right)\check{x}^\top J\check{x}$. By the definitions of the elements of the second conditions, $\check{x}, \lambda^*, p, q$ are all constants that are determined by the SBM and are not dependent on any instance of graph G and J is a constant matrix. Only the matrix D^* is associated with and dependent on each instance of the graph G. Let $D^*, D^{*\prime}$ be the matrices that are associated with G and G', we have to prove that: $\check{x}^\top D^{*\prime}\check{x} + \mathcal{J}(\check{x}) > c_2'\log n$ for the graph G'.

We have that:

$$\begin{aligned}
\check{x}^\top D^{*\prime}\check{x} + \mathcal{J}(\check{x}) &= \left(\check{x}^\top D^{*\prime}\check{x} - \check{x}^\top D^*\check{x}\right) + \left(\check{x}^\top D^*\check{x} + \mathcal{J}(\check{x})\right) && (126)\\
&> \check{x}^\top(D^{*\prime} - D^*)\check{x} + c_2\log n. && (127)
\end{aligned}$$

Now we analyze $\check{x}^\top(D^{*\prime} - D^*)\check{x}$. Let $\Delta^* = D^{*\prime} - D^*$, we have that Δ^* is also a diagonal matrix which $|\Delta_{ii}^*| = |d_i^{*\prime} - d_i^*| = |\sum_j (A_{ij} - A_{ij}')\sigma_i^*\sigma_j^*| \leq \frac{c\log n}{\epsilon}$, since there are at most $\frac{c\log n}{\epsilon}$ different entries between A_{ij} and A_{ij}' for any fixed i. Also, it is similar to check that $|\sum_i \Delta_{ii}^*| \leq 2c\log n/\epsilon$. We set $\mathbf{\Delta} = \{\Delta : |\sum_i \Delta_{ii}| \leq 2c\log n/\epsilon, |\Delta_{ii}| \leq c\log n/\epsilon\}$.

Also, we have:

$$\check{x}^\top (D^{*\prime} - D^*)\check{x} \geq -|\check{x}^\top (D^{*\prime} - D^*)\check{x}| \tag{128}$$

$$\geq -\left| \max_{D^{*\prime\prime}: d(G,G'') \leq c \log n/\epsilon} \check{x}^\top (D^{*\prime\prime} - D^*)\check{x} \right| \tag{129}$$

$$\geq -\left| \max_{\Delta \in \mathbf{\Delta}} \check{x}^\top \Delta \check{x} \right| \tag{130}$$

$$\overset{(a)}{\geq} -\left| \max_{\Delta \in \mathbf{\Delta}} \lambda_{max}(\Delta) \|\check{x}\|_2^2 \right| \tag{131}$$

$$\overset{(b)}{\geq} -\left| \max_{\Delta \in \mathbf{\Delta}} \lambda_{max}(\Delta) \right| \tag{132}$$

$$\overset{(c)}{\geq} -\frac{c \log n}{\epsilon}, \tag{133}$$

where

- (a) is because of $\check{x}^\top \Delta \check{x} \leq \lambda_{max}(\Delta)\|\check{x}\|_2^2$ for any symmetric Δ;
- (b) is because of $\|\check{x}\| = 1$ by its definition;
- and (c) is because of Δ is a diagonal matrix so $\max_\Delta \lambda_{max}(\Delta) = \max_\Delta \max_i \Delta_{ii} \leq c \log n/\epsilon$. Adding this lower bound to Equation 127, we have $\check{x}^\top D^{*\prime} \check{x} + \mathcal{J}(\check{x}) > (c_2 - c/\epsilon) \log n = c_2' \log n$.

The third condition: We will prove that $\|(D^{*\prime} - \mathbb{E}[D^{*\prime}])\check{x}]\|_2 \leq c_3' \sqrt{\log n}$.

We first analyze $\mathbb{E}[D^{*\prime}]$. By the definition of $D^{*\prime}$, $\mathbb{E}[D^{*\prime}] = diag(\mathbb{E}[d^{*\prime}])$. For every i, we have $\mathbb{E}[d^{*\prime}_i] = \mathbb{E}[\sum_j A_{ij}\sigma_i^*\sigma_j^* - \lambda^*(2K-n)\sigma_i^*]$. By the linearity of expectation, we have:

$$\mathbb{E}[d^{*\prime}_i] = \sum_j \mathbb{E}[A'_{ij}]\sigma_i^*\sigma_j^* - \lambda^*(2K-n)\sigma_i^* \tag{134}$$

$$= \sum_j \mathbb{E}[A_{ij}]\sigma_i^*\sigma_j^* - \lambda^*(2K-n)\sigma_i^* \tag{135}$$

$$= \mathbb{E}[d_i^*], \tag{136}$$

since we assume that all graphs G and G' are generated by the same SBM so each entry of their adjacency matrices must have the same expected value, and all other elements of the equations above are constants that only depend on the SBM's parameters. It follows that $\mathbb{E}[D^{*\prime}] = \mathbb{E}[D^*]$.

Applying the property to $(D^{*\prime} - \mathbb{E}[D^{*\prime}])$:

$$\|(D^{*\prime} - \mathbb{E}[D^{*\prime}])\check{x}]\|_2 = \|(D^{*\prime} - D^* + D^* - \mathbb{E}[D^*)\check{x}\|_2 \tag{137}$$

$$\leq \|(D^{*\prime} - D^*)\check{x}\|_2 + \|(D^* - \mathbb{E}[D^*)\check{x}\|_2 \tag{138}$$

$$\leq \|(D^{*\prime} - D^*)\check{x}\|_2 + c_3\sqrt{\log n} \tag{139}$$

For the quantity $\|(D^{*\prime} - D^*)\check{x}\|_2$, using a similar analysis as in Lemma 11, we have;

$$\|(D^{*\prime} - D^*)\check{x}\|_2 = \sqrt{\sum_i \check{x}_i^2 \left(\sum_j (A_{ij} - A'_{ij})\sigma_j^*\right)^2} \tag{140}$$

$$\leq \sqrt{\sum_i \left(\sum_j (A_{ij} - A'_{ij})\sigma_j^*\right)^2} \max\left\{\sqrt{\frac{n-K}{nK}}, \sqrt{\frac{K}{n(n-K)}}\right\} \tag{141}$$

$$\leq \|A - A'\|_F \max\left\{\sqrt{\frac{1-\rho}{\rho}}, \sqrt{\frac{\rho}{1-\rho}}\right\} \tag{142}$$

$$\leq \sqrt{\frac{2c\log n}{\epsilon}}\sqrt{\frac{1-\rho}{\rho}} \tag{143}$$

$$= \sqrt{\frac{2c(1-\rho)}{\epsilon\rho}}\sqrt{logn}. \tag{144}$$

Substituting it to Equation 139, we have:

$$\|(D^{*\prime} - \mathbb{E}[D^{*\prime}])\check{x}]\|_2 \leq \sqrt{\frac{2c(1-\rho)}{\epsilon\rho}}\sqrt{logn} + c_3\sqrt{\log n} \tag{145}$$

$$\leq \left(\sqrt{\frac{2c(1-\rho)}{\epsilon\rho}} + c_3\right)\sqrt{\log n} \tag{146}$$

$$= c'_3\sqrt{\log n} \tag{147}$$

The fourth condition: For any graph G', we prove that $d^{*\prime}_i \geq c'_4 \log n$ for every $i \in [n]$. By the definition of $d^{*\prime}$, we have $\forall i$:

$$d^{*\prime}_i = \sum_j A'_{ij}\sigma_i^*\sigma_j^* - \lambda^*(2K-n)\sigma_i^* \tag{148}$$

$$= \sum_j A'_{ij}\sigma_i^*\sigma_j^* - \sum_j A_{ij}\sigma_i^*\sigma_j^* + \sum_j A_{ij}\sigma_i^*\sigma_j^* - \lambda^*(2K-n)\sigma_i^* \tag{149}$$

$$= \sum_j A'_{ij}\sigma_i^*\sigma_j^* - \sum_j A_{ij}\sigma_i^*\sigma_j^* + d_i^* \tag{150}$$

$$\geq \sum_j (A'_{ij} - A_{ij})\sigma_i^*\sigma_j^* + c_4\log n \tag{151}$$

$$\overset{(a)}{\geq} -\frac{c\log n}{\epsilon} + c_4\log n \tag{152}$$

$$= \left(-\frac{c}{\epsilon} + c_4\right)\log n \tag{153}$$

$$= c'_4\log n, \tag{154}$$

where (a) is because for each fixed i, A_{ij}s and A'_{ij}s differ by at most $c\log n/\epsilon$ entries across all j.

□

Lemma 16. *(Lemma 3 of (Hajek et al., 2016b)) Suppose there exist $D^* = diag\{d_i^*\}$ and $\lambda^* \in \mathbb{R}$ such that $S^* = D^* - A + \lambda^* J$ satisfies $S^* \succcurlyeq 0, \lambda_2(S^*) > 0$ and $S^*\sigma^* = 0$. Then Y^* is the unique solution of the program $SDP(G)$.*

Lemma 17. *If G is $\mathcal{C}$-concentrated then $SDP(G) = Y^*$, i.e, the optimal solution of $SDP(G)$ is the ground truth community matrix $Y^* = \sigma^* \sigma^{*T}$.*

Proof. With D^* and λ^* defined as above, we show that when a graph G is $\mathcal{C}$-concentrated, $S^* = D^* - A + \lambda^* J$ satisfies Lemma 16's requirements and the Lemma follows.

We recall that

- $\tau = \frac{a-b}{\log a - \log b}$
- $\lambda^* = \tau \log n / n$
- $d^* \in \mathbb{R}^n : d_i^* = \sum_{j=1}^n A_{ij} \sigma_i^* \sigma_j^* - \lambda^* (2K - n) \sigma_i^*$
- $D^* = diag\{d^*\}$

For any i, we have $\sigma_i^* \sigma_i^* = 1$. Therefore, $d_i^* \sigma_i^* = \sum_{j=1}^n A_{ij} \sigma_i^* \sigma_j^* \sigma_i^* - \lambda^* (2K-n) \sigma_i^* \sigma_i^* = \sum_{j=1}^n A_{ij} \sigma_j^* - \lambda^* (2K-n)$. We have $D^* \sigma^* = A\sigma^* - \lambda^* (2K-n)\mathbf{1}$. It follows that $S^* \sigma^* = D^* \sigma^* - A\sigma^* + \lambda^* J \sigma^* = A\sigma^* - \lambda^* (2K-n)\mathbf{1} - A\sigma^* - \lambda^* J \sigma^* = 0$, which satisfies the condition that $S^* \sigma^* = 0$. Because of this, proving $\inf_{x \perp \sigma^*, \|x\|=1} x^T S^* x > 0$ is sufficient to satisfy all remaining conditions, since all feasible x plus σ^* will include a basis for the whole space, which means $\forall y : y^T S^* y \geq 0$ ($S^* \succcurlyeq 0$) and the solution set of $S^* y = 0$ has only 1 dimension ($\lambda_2(S^*) > 0$).

From now, we will show that when a graph G is $\mathcal{C}$-concentrated,

$$\inf_{x \perp \sigma^*, \|x\|=1} x^T S^* x > 0 \tag{155}$$

with probability 1.

Note that $\mathbb{E}[A] = \frac{p-q}{2} Y^* + \frac{p+q}{2} J - pI$. For any $x : x \perp \sigma^*, \|x\| = 1$:

$$\begin{aligned}
x^T S^* x &= x^T D^* x - x^T A x + x^T \lambda^* J x && (156)\\
&= x^T D^* x - x^T (A - \mathbb{E}[A]) x - x^T \mathbb{E}[A] x + x^T \lambda^* J x && (157)\\
&= x^T D^* x - x^T (A - \mathbb{E}[A]) x - \frac{p-q}{2} x^T Y^* x - \frac{p+q}{2} x^T J x + p x^T I x - \lambda^* x^T J x && (158)\\
&= x^T D^* x - x^T (A - \mathbb{E}[A]) x - \frac{p-q}{2} x^T \sigma^* \sigma^{*T} x - \frac{p+q}{2} x^T J x + p x^T I x - \lambda^* x^T J x && (159)\\
&\overset{(a)}{=} x^T D^* x - x^T (A - \mathbb{E}[A]) x + (\lambda^* - \frac{p+q}{2}) x^T J x + p \|x\|_2 && (160)\\
&= x^T D^* x - x^T (A - \mathbb{E}[A]) x + (\lambda^* - \frac{p+q}{2}) x^T J x + p && (161)\\
&\overset{(b)}{\geq} p - \lambda_1 (A - \mathbb{E}[A]) \|x\|_2^2 + x^T D^* x + (\lambda^* - \frac{p+q}{2}) x^T J x && (162)\\
&= p - \lambda_1 (A - \mathbb{E}[A]) + x^T D^* x + (\lambda^* - \frac{p+q}{2}) x^T J x && (163)\\
&\overset{(c)}{\geq} p - \|A - \mathbb{E}[A]\|_2 + x^T D^* x + (\lambda^* - \frac{p+q}{2}) x^T J x && (164)\\
&\overset{(d)}{\geq} p - c_1 \sqrt{\log n} + x^T D^* x + (\lambda^* - \frac{p+q}{2}) x^T J x, && (165)
\end{aligned}$$

where:

- (a) is because $x^T\sigma^* = 0$,
- (b) is because $x^T Bx < \lambda_1(B)\|x\|_2^2$ for any symmetric matrix B,
- (c) is because $\lambda(B) < \|B\|_2$ for any matrix B,
- and (d) is because of the **First condition** of $\mathcal{C}$-concentration.

Let $t(x) = x^T D^* x + (\lambda^* - \frac{p+q}{2})x^T Jx$. By the definition of $\check{x}$ (Definition 4), we define $E = span(\check{x}, \sigma^*)$. Any $y : y \perp \sigma^*, \|y\|_2 = 1$ can be represented as $y = \beta\check{x} + \sqrt{1-\beta^2}x$ for $x \in \{x : x \perp E, \|x\|_2 = 1\}$ and $\beta \in [0,1]$. We have:

$$\inf_{x\perp\sigma^*,\|x\|_2=1} t(x) = \inf_{x\perp E,\|x\|_2=1,\beta\in[0,1]} t(\beta\check{x} + \sqrt{1-\beta^2}x) \tag{166}$$

$$\overset{(a)}{=} \inf_{x\perp E,\|x\|_2=1,\beta\in[0,1]} \left(\beta^2(\check{x}^T D^*\check{x} + \mathcal{J}(\check{x})) + 2\beta\sqrt{1-\beta^2}x^T D^*\check{x} + (1-\beta^2)x^T D^* x\right) \tag{167}$$

$$\geq \inf_{\beta\in 0,1}(\beta^2(\check{x}^T D^*\check{x} + \mathcal{J}(\check{x})) + \inf_{x\perp E,\|x\|_2=1,\beta\in[0,1]}(2\beta\sqrt{1-\beta^2}x^T D^*\check{x} + (1-\beta^2)x^T D^* x) \tag{168}$$

$$\overset{(b)}{\geq} \inf_{\beta\in 0,1}(\beta^2 c_2\log n + \inf_{x\perp E,\|x\|_2=1,\beta\in[0,1]}(2\beta\sqrt{1-\beta^2}x^T D^*\check{x} + (1-\beta^2)x^T D^* x) \tag{169}$$

$$\overset{(c)}{\geq} \inf_{\beta\in 0,1}(\beta^2 c_2\log n - 2\beta\sqrt{1-\beta^2}\|(D^* - \mathbb{E}[D^*])\check{x})\| + \inf_{x\perp E,\|x\|_2=1,\beta\in[0,1]}(1-\beta^2)x^T D^* x) \tag{170}$$

$$\overset{(d)}{\geq} \inf_{\beta\in 0,1}(\beta^2 c_2\log n - 2\beta\sqrt{1-\beta^2}c_3\sqrt{\log n} + \inf_{x\perp E,\|x\|_2=1,\beta\in[0,1]}(1-\beta^2)x^T D^* x) \tag{171}$$

$$\overset{(e)}{\geq} \inf_{\beta\in 0,1}(\beta^2 c_2\log n - 2\beta\sqrt{1-\beta^2}c_3\sqrt{\log n} + (1-\beta^2)\min_i d_i^*) \tag{172}$$

$$\overset{(f)}{\geq} \inf_{\beta\in 0,1}(\beta^2 c_2\log n - 2\beta\sqrt{1-\beta^2}c_3\sqrt{\log n} + (1-\beta^2)c_4\log n) \tag{173}$$

$$\geq \inf_{\beta\in 0,1}(\beta^2 c_2\log n + (1-\beta^2)c_4\log n) - c_3\sqrt{\log n} \tag{174}$$

$$\geq \frac{1}{2}\min\{c_2, c_4\}\log n - c_3\sqrt{\log n}. \tag{175}$$

where:

- (a) is because $Jx = 0$,
- (b) is because $\check{x}^T D^*\check{x} + \mathcal{J}(\check{x}) \geq c_2\log n$ as the **second condition**,
- (c) is because $\inf_{x\perp E,\|x\|_2=1} x^T D^*\check{x} = \inf_{x\perp E,\|x\|_2=1} x^T(D^* - \mathbb{E}[D^*])\check{x} \geq -\|(D^* - \mathbb{E}[D^*])\|$,
- (d) is because $\|(D^* - \mathbb{E}[D^*])\| < c_3\sqrt{\log n}$ as the **third condition**,
- (e) is because D^* is a diagonal matrix constructed from d_i^*,
- (f) is because $min_i d_i^* \geq c_4\log n$ as the **fourth condition**.

Substituting the lower bound of $t(x)$ to Equation 165, we have:

$$x^T S^* x \geq \frac{1}{2}\min\{c_2, c_4\}\log n - (c_1 + c_3)\sqrt{\log n} + p \tag{176}$$

$$> 0, \tag{177}$$

where n is large enough.

□

Lemma 18. *A graph G generated by a Binary Asymmetric SBM with two communities sized ρn and $(1-\rho)n$ for some constant ρ and with $p = \frac{a\log n}{n}$ and $q = \frac{b\log n}{n}$ is $\frac{c}{\epsilon}\log n$-stable under $SDP(G)$ with probability at least $1 - n^{-\Omega(1)}$*

Proof. By Lemma 14, G is $\mathcal{C}$-concentrated with probability at least $1 - n^{-\Omega(1)}$. By Lemma 15, all graphs G' (which include G itself) with distance up to $\frac{c}{\epsilon}\log n$ from G are also $\mathcal{C}'$-concentrated. By Lemma 17, since all graphs G' are C'-concentrated, $SDP(G')$ always output the optimal and unique solution σ^* with probability 1, or $SDP(G) = SDP(G') = \sigma^*$ for all G'. It follows that G is $\frac{c}{\epsilon}\log n$-stable under SDP with probability at least $1 - n^{-\Omega(1)}$. □

Theorem 6. *Given a graph G generated by a Binary Asymmetric SBM with two communities sized ρn and $(1-\rho)n$ for some constant ρ, and with $p = \frac{a\log n}{n}$ and $q = \frac{b\log n}{n}$, $\tilde{h}(c/\epsilon) > 1$, $\sqrt{a} - \sqrt{b(1+\log\frac{a}{b})} > \sqrt{c\log\frac{a}{b}/\epsilon}$, M_{SDP} with $\delta = n^{-\Omega(1)}$ exactly recovers the ground-truth community σ^*, i.e., $\Pr[M^{SDP}(G) \neq \sigma^*] = n^{-\Omega(1)}$*

Proof. Lemma 18 states the stability property of G under SDP, i.e., G is $\frac{c}{\epsilon}\log n$-stable under SDP. It also implies that $SDP(G) = \sigma^*$ under the effect of $\mathcal{C}$-concentration. By applying Lemma 6, $\Pr[M^{SDP}(G) \neq SDP(G)] = n^{-\Omega(1)}$ or $\Pr[M^{SDP}(G) \neq \sigma^*] = n^{-\Omega(1)}$.

The condition for the Theorem is that $\mathcal{C}'$ is a valid constant combination, i.e., $c_2' > 0$ and $c_4' > 0$, or $c_2 - c/\epsilon > 0$ and $c_4 - c/\epsilon > 0$ with pre-determined c, ϵ.

Lemma 13 requires that $\tilde{h}(c_4) > 1$, which means $\tilde{h}(c/\epsilon) > 1$

Lemma 10 requires that $c_2 < \tau - b$, which means $c/\epsilon < \tau - b$. This is equivalent to $a - b(1+\log\frac{a}{b}) > c\log\frac{a}{b}/\epsilon$, which can be simplified to $\sqrt{a} - \sqrt{b(1+\log\frac{a}{b})} > \sqrt{c\log\frac{a}{b}/\epsilon}$. □

C. Censored Binary SBM (CBSBM)

In this section we analyze the Stability mechanism on the recoverability of the Censored Binary SBM. In this model, the generated graphs are bedge-weighted. The vertices are consists of nodes from two clusters with possibly unequal sizes. Edges between these nodes are generated by an Erdos-Renyi model $G(n,p)$, regardless of the endpoint's communities. With a fixed constant $\xi \in [0, 0.5]$, each edge (i,j) has a label $L_{ij} \in \{+1,-1\}$ drawn from the following distribution:

Definition 7. *Definition of CBSBM.*

$$P_{L_{ij}|\sigma_i^*,\sigma_j^*} = (1-\xi)\mathbf{1}_{L_{ij}=\sigma_i^*\sigma_j^*} + \xi\mathbf{1}_{L_{ij}=-\sigma_i^*\sigma_j^*} \tag{178}$$

Privacy model. The adjacency matrix $A(G)$ is defined as $A_{ij} = 0$ if there is no edge between i and j. $A_{ij} = L_{ij}$ if there is an edge generated between i and j. In this section, we define the neighborhood between two graph $G \sim G'$ if $A(G) and A(G')$ differ by exact one element. This privacy model can protect the existence (and the non-existence) of an arbitrary edge (i,j), where any two neighboring adjacency matrices differ at element ij: $A(G)_{ij} = 0$ (not an edge) and $A(G')_{ij} \neq 0$ (an edge). It can also protect the label of an arbitrary edge (i,j) whenever (i,j) exists in the input graphs, that any two neighboring adjacency matrix differ as follows: $A(G)_{ij} = -1$ and $A(G')_{ij} = +1$.

In the non-private setting, communities in the CBSBM can be recovered exactly by solving the following SDP Relaxation:

Definition 8. *SDP Relaxation of the Censored Binary SBM:*

$$\hat{Y}_{SDP} = arg\ max \langle A, Y\rangle \tag{179}$$

$$s.t.\ Y \succcurlyeq 0 \tag{180}$$

$$Y_{ii} = 1, i \in [n]. \tag{181}$$

Assumptions of parameters. We assume that $p = a\log n/n$ for some fixed constant a (in the random edge generation model $G(n,p)$). For the random label model: $h(\xi, a) = a(\sqrt{1-\xi} - \sqrt{\xi})^2 > 1$, or $h(\xi, a) = 1 + \Omega(1)$. We may drop parameter a when the context is clear (CBSBM with a fixed a).

In this section, we denote the $SDP(G)$ as a function taking input graph G and outputting the optimal solution of the SDP Relaxation constructed by its adjacency matrix $A(G)$.

Definition 9. *In this section, we define the following quantities:*

- $d_i^* = \sum_{j=1}^n A_{ij}\sigma_i^*\sigma_j^*$ *for every* $i \in [n]$
- $D^* = diag\{d^*\}$

Definition 10. *Definition of C-concentration. Assume a graph G is generated by a CBSBM with two communities with ground truth vector σ^*, edge probability $p = \frac{a\log n}{n}$, and edge labels are generated by the above process with some constant $\xi \in [0, 0.5]$.*

G is called $\mathcal{C}$-concentrated if there exists a tuple (c_1, c_2) such that G satisfies two conditions

- $\|A - \mathbb{E}[A]\|_2 \leq c_1\sqrt{\log n}$
- $d_i^* \geq c_2 \log n$ *for every* $i \in [n]$

Lemma 19. *(Theorem 9 of(Hajek et al., 2016b)) Given a graph G generated by a CBSBM as in Definition 7, there exists a constant $c_1 > 0$ such that with probability at least $1 - n^{-\Omega(1)}$, we have:*

$$\|A - \mathbb{E}[A]\|_2 \leq c_1\sqrt{\log n} \tag{182}$$

Lemma 20. *(Lemma 1 of (Hajek et al., 2016b)) Let $X \sim Binom(m, a\log n/n)$ for $m \in \mathbb{N}$ where $m = \rho n + o(n)$ for some $\rho > 0$ as $n \to \infty$. Let $k_n \in [m]$ be such that $k_n = \tau\rho\log n + o(\log n)$ for some $0 \leq \tau \leq a$, then:*

$$\Pr[X \leq k_n] = n^{\rho(a - \tau\log\frac{ea}{\tau} + o(1))}. \tag{183}$$

Lemma 21. *Given a graph G generated by a CBSBM as in Definition 7, there exists a constant $0 < c_2 < a$ such that with probability at least $1 - n^{-\Omega(1)}$, we have:*

$$\min_{i\in[n]} d_i^* \geq c_2 \log n \tag{184}$$

Proof. The proof here is inspired by (Hajek et al., 2016b), with the main differences are the bound in the original proof is in order of $\frac{\log n}{\log\log n}$, where we need a bound in order of $\log n$ for the remaining proof of Stability to work. By definition, $d_i^* = \sum_{j=1}^n A_{ij}\sigma_i^*\sigma_j^*$. In the generation process, it is equivalent to $d_i^* = \sum_{j=1}^{n-1} X_j$, where $X_j \overset{i.i.d.}{\sim} p(1-\xi)\beta_{+1} + p\xi\beta_{-1} + (1-p)\beta_0$ where β_x is the Dirac delta function at x. Hence, we will prove that with probability at least $1 - n^{-\Omega(1)}$, there exist some constant $c_2 > 0$ that for every i:

$$\Pr[\sum_{j=1}^n X_j < c_2 \log n] \leq n^{-\Omega(1)} \tag{185}$$

We first analyze the case when $\xi = 0$. Then $\sum_{j=1}^n X_j \sim Binom(n, a\log n/n)$. It follows that:

$$\begin{aligned}
\Pr[\sum_{j=1}^n X_j < c_2 \log n] &\leq n^{-\rho(a - \tau\log\frac{ea}{\tau} + o(1))} \text{ by Lemma 20} && (186)\\
&= n^{-a + c_2\log\frac{ea}{c_2} - o(1)} \text{ by substituting } \rho = 1, \tau = c_2 && (187)\\
&\leq n^{-a - o(1)} \text{ since } c_2 < a && (188)\\
&\leq n^{-h(\xi)} \text{ since } h(\xi) = a \text{ when } \xi = 0 && (189)\\
&\leq n^{-1-\Omega(1)}. && (190)
\end{aligned}$$

Taking the union bound over all nodes i, we have $d_i^* \geq c_2 \log n$ with probability at least $1 - n^{-\Omega(1)}$.

Now we analyze the case when $\xi > 0$. By the Chernoff bound:

$$\Pr[\sum_{j=1}^{n} X_j < c_2 \log n] \leq \exp(-n\ell(\frac{c_2 \log n}{n})), \tag{191}$$

where the function $\ell(x)$ is defined as $\ell(x) = \sup_{\lambda \geq 0} -\lambda x - \log \mathbb{E}[e^{-\lambda X}]$ with $X \sim p(1-\xi)\beta_{+1} + p\xi\beta_{-1} + (1-p)\beta_0$, or $\mathbb{E}[e^{-\lambda X}] = 1 + p(e^{-\lambda}(1-\xi) + e^{\lambda}\xi - 1)$. Since $\ell(x)$ is concave (in λ), the supremum at λ^* is:

$$-x + \frac{p(e^{-\lambda^*}(1-\xi) - e^{\lambda^*}\xi}{1 + p(e^{-\lambda^*}(1-\xi) + e^{\lambda^*}\xi - 1)} = 0. \tag{192}$$

Substituting $x = \frac{c_2 \log n}{n} \approx 0$, we solve λ^* as follows:

$$\begin{aligned}
e^{-\lambda^*}(1-\xi) - e^{\lambda^*}\xi &= o(1) && (193)\\
\implies e^{\lambda^*}(e^{-2\lambda^*}(1-\xi) - xi) &= o(1) && (194)\\
\implies e^{-2\lambda^*} &= \frac{\xi}{1-\xi} + o(1) && (195)\\
\implies -2\lambda^* &= \log \frac{\xi}{1-\xi} + o(1) && (196)\\
\implies \lambda^* &= \frac{1}{2} \log \frac{1-\xi}{\xi} + o(1). && (197)
\end{aligned}$$

Substituting $x = \frac{c_2 \log n}{n}$ and $\lambda^* = \frac{1}{2} \log \frac{1-\xi}{\xi} + o(1)$, we have:

$$\begin{aligned}
\ell(\frac{c_2 \log n}{n}) &= -\lambda^* \frac{c_2 \log n}{n} - \log(1 + p(e^{-\lambda^*}(1-\xi) + e^{\lambda^*}\xi - 1)) && (198)\\
&= -\frac{1}{2} \log \frac{1-\xi}{\xi} \times \frac{c_2 \log n}{n} - \log(1 - p(\sqrt{1-\xi} - \sqrt{\xi})^2) + o(\frac{\log n}{n}) && (199)\\
&\lessapprox -\log(1 - p(\sqrt{1-\xi} - \sqrt{\xi})^2) + o(\frac{\log n}{n}), \text{ since } \frac{c_2 \log n}{n} \gtrapprox 0 && (200)\\
&= -p(\sqrt{1-\xi} - \sqrt{\xi})^2 + o(\frac{\log n}{n}), \text{ since } \log(1-x) = -x \text{ at } x \approx 0 && (201)\\
&= \frac{a \log n}{n}(\sqrt{1-\xi} - \sqrt{\xi})^2 + o(\frac{\log n}{n}). && (202)
\end{aligned}$$

Applying the above result to Equation 191, we have:

$$\begin{aligned}
\Pr[\sum_{j=1}^{n} X_j < c_2 \log n] &\leq \exp(-n\ell(\frac{c_2 \log n}{n})) && (203)\\
&\leq \exp(-n \times \frac{a \log n}{n}(\sqrt{1-\xi} - \sqrt{\xi})^2 - n \times o(\frac{\log n}{n})) && (204)\\
&\leq \exp(-a \log n(\sqrt{1-\xi} - \sqrt{\xi})^2 - o(\log n)) && (205)\\
&\leq n^{-h(\xi) - \Omega(1)} && (206)\\
&\leq n^{-1-\Omega(1)}. && (207)
\end{aligned}$$

Taking the union bound over all nodes i, we have $d_i^* \geq c_2 \log n$ with probability at least $1 - n^{-\Omega(1)}$ and the Lemma follows. □

Lemma 22. *A graph G generated by a CBSBM as in Definition 7, there exists some tuple of constants (c_1, c_2) such that with probability at least $1 - n^{-\Omega(1)}$, G is (c_1, c_2)-concentrated ($\mathcal{C}$-concentrated).*

Proof. Refer to the definition of $\mathcal{C}$-concentrated for CBSBM (Definition 10), the **First condition** follows from Lemma 19 and the **Second condition** follows from Lemma 21. Applying union bound on both conditions, we have that with probability at least $1 - n^{-\Omega(1)}$, G is $\mathcal{C}$-concentrated. □

Lemma 23. *Assume a fixed CBSBM models, if a graph G generated by a CBSBM is (c_1, c_2)-concentrated then every graph G' within distance $\frac{c \log n}{\epsilon}$ of G, i.e. $d(G, G') < \frac{c \log n}{\epsilon}$, is (c_1', c_2')-concentrated where $c_1' = c_1 + \sqrt{8c/\epsilon}$, $c_2' = c_2 - c/\epsilon$.*

Proof. **The first condition:** It is clear that $A(G')$ can be obtained by changing up to $\frac{c \log n}{\epsilon}$ cells of $A(G)$, with the largest impact (to the concentration properties) by changing some cell ij from $A(Gx)_{ij} = 1$ to $A(G')_{ij} = -1$ or vice versa. It follows that $\|A(G) - A(G')\|_F^2 \leq \frac{8c \log n}{\epsilon}$. Also, since we assume G and G' are under the same CBSBM, $\mathbb{E}[A(G')] = \mathbb{E}[A(G)] = \bar{A}$.

Now we analyze $\|A(G') - \mathbb{E}[A(G')]\|_2$:

$$\begin{aligned}
\|A(G') - \mathbb{E}[A(G')]\|_2 &= \|A(G') - \bar{A}\|_2 && (208)\\
&= \|A' - A + A - \bar{A}\|_2 && (209)\\
&\leq \|A' - A\|_2 + \|A - \bar{A}\|_2 && (210)\\
&\leq \|A' - A\|_2 + c_1\sqrt{\log n} && (211)\\
&\leq \sqrt{\frac{8c \log n}{\epsilon}} + c_1\sqrt{\log n} && (212)\\
&= (\sqrt{8c/\epsilon} + c_1)\sqrt{\log n} && (213)\\
&= c_1'\sqrt{\log n}, && (214)
\end{aligned}$$

which means that G' satisfies the **First condition** with some constant c_1'.

The first condition: By the definition of $d_i'^*$, we have:

$$\begin{aligned}
d_i'^* &= \sum_{j=1}^{n} A'_{ij}\sigma_i^*\sigma_j^* && (215)\\
&= \sum_{j=1}^{n}(A'_{ij} - A_{ij})\sigma_i^*\sigma_j^* + \sum_{j=1}^{n} A_{ij}\sigma_i^*\sigma_j^* && (216)\\
&\geq \sum_{j=1}^{n}(A'_{ij} - A_{ij})\sigma_i^*\sigma_j^* + c_2 \log n && (217)\\
&\geq -\frac{2c \log n}{\epsilon} + c_2 \log n && (218)\\
&= (c_2 - \frac{2c}{\epsilon}) \log n && (219)\\
&= c_2' \log n, && (220)
\end{aligned}$$

which means that G' satisfies the **Second condition** with some constant c_2', and the Lemma follows. □

Lemma 24. *(Lemma 9 of(Hajek et al., 2016b)) Suppose there exists $D^* = diag d_i^*$ such that $S^* = D^* - A$ satisfies $S^* \succcurlyeq 0, \lambda_2(S^*) > 0$ and $S^*\sigma^* = 0$. Then Y^* is the unique solution of the program $SDP(G)$.*

Lemma 25. *If G is $\mathcal{C}$-concentrated then $SDP(G) = Y^*$, i.e., the optimal solution of $SDP(G)$ is the ground truth community matrix $Y^* = sigma^*\sigma^{*T}$.*

Proof. Constructing $d_i^* = \sum_{j=1}^n A_{ij}\sigma_i^*\sigma_j^*$, we will prove that S^* satisfies the condition of Lemma 24 and due to Lemma 24, the Lemma follows that Y^* is the optimal solution of $SDP(G)$.

First, we prove that $S^*\sigma^* = 0$. By definition of $S^* = D^* - A$, $S^*\sigma^* = D^*\sigma^* - A\sigma^* = A^*\sigma^* - A^*\sigma^* = 0$

Second, proving that $\inf_{x\perp\sigma^*,\|x\|=1} x^T S^* x > 0$ satisfies other conditions since all feasible x plus σ^* will include a basis for the whole space, which means $\forall y : y^T S^* y \geq 0$, or $S^* \succcurlyeq 0$ and the solution set of $S^* y = 0$ has only 1 dimension $\lambda_2(S^*) > 0$.

Therefore, the next step is to prove that when G is $\mathcal{C}$-concentrated, $\inf_{x\perp\sigma^*,\|x\|=1} x^T S^* x > 0$ with probability 1 and the proof follows. For any $x \perp \sigma^*$, $\|x\| = 1$, we have:

$$
\begin{aligned}
x^T S^* x &= x^T D^* x - x^T A x && (221)\\
&= x^T D^* x - x^T (A - \mathbb{E}[A])x - x^T \mathbb{E}[A]x && (222)\\
&\overset{(a)}{\geq} \min_{i\in[n]} d_i^* - \|A - \mathbb{E}[A]\|_2 - x^T \mathbb{E}[A]x && (223)\\
&\overset{(b)}{\geq} c_2 \log n - c_1\sqrt{\log n} - x^T \mathbb{E}[A]x && (224)\\
&\overset{(c)}{=} c_2 \log n - c_1\sqrt{\log n} + (1-2\xi)p x^T Y^* x && (225)\\
&= c_2 \log n - c_1\sqrt{\log n} + (1-2\xi)p && (226)\\
&= c_2 \log n - c_1\sqrt{\log n} + (1-2\xi)a\log n/n && (227)\\
&\geq 0 \text{ where } n \text{ is large enough,} && (228)
\end{aligned}
$$

where:

- (a) is because of $\|x\| = 1$,
- (b) is because of the $\mathcal{C}$-concentration's conditions,
- (c) is because of $\mathbb{E}[A] = (1-2\xi)Y^*$.

□

Lemma 26. *A graph G generated by a CBSBM as in Definition 7 is $\frac{c}{\epsilon}\log n$-stable under $SDP(G)$ with probability at least $1 - n^{-\Omega(1)}$.*

Proof. By Lemma 22, G is $\mathcal{C}$-concentrated with probability at least $1 - n^{-\Omega(1)}$. By Lemma 23, all graphs G' (which include G itself) with distance up to $\frac{c}{\epsilon}\log n$ from G are also $\mathcal{C}'$-concentrated. By Lemma 25, since all graphs G' are C'-concentrated, $SDP(G')$ always output the optimal and unique solution σ^* with probability 1, or $SDP(G) = SDP(G') = Y^*$ for all G'. It follows that G is $\frac{c}{\epsilon}\log n$-stable under SDP with probability at least $1 - n^{-\Omega(1)}$. □

Theorem 7. *Given graph G generated by a CBSBM as in Definition 7 , and $c/\epsilon < a$ and $h(\xi, a) > 1, c/\epsilon < a$, M_{SDP} with $\delta = n^{-\Omega(1)}$ exactly recovers the ground-truth community Y^*, i.e., $\Pr[M^{SDP}(G) \neq Y^*] = n^{-\Omega(1)}$*

Proof. Lemma 26 states the stability property of G under SDP, i.e., G is $\frac{c}{\epsilon}\log n$-stable under SDP. It also implies that $SDP(G) = Y^*$ under the effect of $\mathcal{C}$-concentration. By applying Lemma 6, $\Pr[M^{SDP}(G) \neq SDP(G)] = n^{-\Omega(1)}$ or $\Pr[M^{SDP}(G) \neq Y^*] = n^{-\Omega(1)}$.

The condition for the Theorem is that $\mathcal{C}'$ is a valid constant combination, i.e., $c_2' < a$ and hence, $h(\xi, a) > 1$ and $c/\epsilon < a$. □

D. Polynomial Algorithms

In Algorithm 1, there are two computation tasks: computing $d_{SDP}(G)$ and $SDP(G)$. Solving the SDP Relaxation can be done in polynomial-time. Therefore, the remaining question is how long it takes to compute $d_{SDG}(G)$. One way is to start with all neighbors G' of distance $k = 1$ of G and test if $SDP(G) \stackrel{?}{=} SDP(G')$. If all G' satisfy the test, we can conclude that G is at least 1-stable, otherwise G is unstable. If G is unstable, we increase k by one and repeat the procedure until we find the first G'' that has $SDP(G'') \neq SDP(G)$ and $d_{SDP}(G) = k$. We may apply the trick of (Mohamed et al., 2022) that stops when $k = c \log n/\epsilon$. In this case, because there are $n^{O(\log n)}$ neighbors of distance up to $c \log n/\epsilon$ from G, we have to invoke the solver for the SDP Relaxation $n^{O(\log n)}$ times. Since $d_{SDP}(G) = O(\log n)$ w.h.p., Algorithm 1 takes $n^{O(\log n)}$ w.h.p..

The main idea is if we can estimate $d(G)$ faster, we can design a faster algorithm. We note that if the input graph is $\mathcal{C}$-concentrated, then $d(G) \geq c \log n/\epsilon$. Therefore, we can test if the input graph is $\mathcal{C}$-concentrated. If the test comes out as positive, we can set $\widehat{d}(G) = d(G) \geq c \log n/\epsilon$ and use $\widehat{d}$ instead of d. Else, we compute $\widehat{d}(G) = \min(d(G), c \log n/\epsilon)$. It is clear that w.h.p., the test is positive. The main challenge is that, testing $\mathcal{C}$-concentration requires knowledge of the SBMs, i.e., p, q and **most importantly**, σ^* (or ξ_k^* in $r > 2$ communities)–the quantities we are trying to output.

In several settings of applications, the edge probabilities p and q (and therefore a, b respectively) may be known, which makes the problem easier. Generally, it is safe to assume that we do not know a and b. To construct the conditions of $\mathcal{C}$-concentration, we need a reliable way to estimate them from the input graph.

Suppose that we have access to oracles that can provide us these quantities. Let $\textsc{Oracle}_{\sigma^*}$ be the one that can provide us the true value of σ^* (or ξ_k^* respectively). Let $\textsc{Oracle}_{a,b}^{\alpha}$ be the one that can provide us the parameters $\hat{a}, \hat{b}$ accurately up to a factor of $1 \pm \alpha$ from the true values of a, b for a small constant $\alpha < 0.001$.

We present Algorithm 3 with the unrealistic assumption of the oracles. We then prove that Algorithm 3 retains the privacy and utility of Algorithm 1, which means it is private and achieve exact recovery. Because checking the $\mathcal{C}$-concentration can be done in polynomial time, and w.h.p., we do not have to invoke high computational cost $d(G)$, the Algorithm takes polynomial time. After we confirm that $\mathcal{M}_{\text{Stbl Oracle}}$ has all the properties we need, we will replace the oracles by realistic components that we calculate from the input graph G. We then present Algorithm 2 that w.h.p. is the same with Algorithm 3.

In order to do that, we employ an estimator derived from the results of (Hajek et al., 2016b) and formalize it in Algorithm 4.

Algorithm 3 $\mathcal{M}^f_{\text{Stbl Oracle}}(G, \mathcal{C})$: Fast Stability Mechanism with Oracle

$\hat{Y} \leftarrow f(G)$
$\sigma^* \leftarrow \textsc{Oracle}_{\sigma^*}$
$(\hat{a}, \hat{b}) \leftarrow \textsc{Oracle}_{a,b}^{\alpha}$
$\hat{\mathcal{C}} \leftarrow$ adjust $\mathcal{C}$ on α to satisfy Proposition 4
Construct $\hat{\mathcal{C}}$-concentration using $\hat{C}, \sigma^*, \hat{a}, \hat{b}$
if G is $\hat{\mathcal{C}}$-concentrated **then**
    $\hat{d}(G) \leftarrow c \log n/\epsilon$
else
    $\hat{d}(G) \leftarrow \min(c \log n/\epsilon, d(G))$
end if
$\tilde{d}(G) \leftarrow \hat{d}(G) + \text{Lap}(1/\epsilon)$
if $\tilde{d}_f(G) > \frac{\log 1/\delta}{\epsilon}$ **then**
    Output $\hat{Y}$
else
    Output $\perp$
end if

In each SBM setting, the proposition can be easily verified by checking all conditions of $\mathcal{C}$-concentration. $\textsc{Oracle}_{\sigma^*}$ guarantees us the true value of σ^*, so the differences between $\mathcal{C}$ and $\hat{\mathcal{C}}$ only come from the factor α of $\textsc{Oracle}_{a,b}^{\alpha}$. We use a tighter tuple of constants $\hat{\mathcal{C}}$ (compared to $\mathcal{C}$) to balance the fact that $\hat{a}$ and $\hat{b}$ may be off by some factor of $1 \pm \alpha$. This task

can be done by adjust each condition by scaling the respective c_k to a factor of $1 \pm 2\alpha$ in which direction that makes the condition tighter.

Lemma 27. *Algorithm 3 is (ϵ, δ)-differentially private.*

Proof. Suppose that $\Delta_{\hat{d}} = 1$ (the global sensitivity of $\hat{d}$). Using the same arguments in the proof of Theorem 5, substituting d by $\hat{d}$, it follows that the algorithm is (ϵ, δ)-differentially private. It remains that we need to prove $\Delta_{\hat{d}} = 1$.

Suppose we have a pair of neighbors $G \sim G'$. By definition $\Delta_{\hat{d}} = \max_{G \sim G'} |\hat{d}(G) - \hat{d}(G')|$. If both of them pass the test in line 6, $\hat{d}(G) = \hat{d}(G') = c \log n/\epsilon$ which means $|\hat{d}(G) - \hat{d}(G')| = 0 < 1$. If both of them fail the test in line 6, $\hat{d}(G) = d(G)$ and $\hat{d}(G') = d(G)$, which means $|\hat{d}(G) - \hat{d}(G')| = |d(G) - d(G')| \leq 1$.

In the last case, assume that G passes the test in line 6 while G' fails. It means that $\hat{d}(G) = c \log n/\epsilon$ and $\hat{d}(G') = d(G')$. Suppose $d(G') \leq c \log n/\epsilon - 2$, because $\Delta_d = 1$ and $G \sim G'$, $d(G) \leq d(G') + 1 \leq c \log n/\epsilon - 1$. But it contradicts with the fact that G is $\hat{\mathcal{C}}$-concentrated which implies that that G is $c \log n/\epsilon$-stable or $d(G) \geq c \log n/\epsilon$. It shows that $d(G') > c \log n/\epsilon - 2$ or $\hat{d}(G') \geq c \log n/\epsilon$ or $\hat{d}(G) - \hat{d}(G') \leq 1$ and the Lemma follows. □

Lemma 28. *If $\mathcal{M}^f_{\text{Stbl}}$ (Algorithm 1) achieves exact recovery (under some specific conditions of the SBMs under the view of Lemma 1), $\mathcal{M}^f_{\text{Stbl ORACLE}}$ (Algorithm 3) also achieves exact recovery under the same conditions.*

Proof. Since $\hat{\mathcal{C}}$-concentration is a valid concentration, it follows that $\Pr[G \text{ is } \hat{\mathcal{C}}\text{-concentrated}] \geq 1 - n^{-\Omega(1)}$. Also, since $\hat{C}$-concentration implies $\mathcal{C}$-concentration by Proposition 4, and all graphs satisfies $\mathcal{C}$-concentration are $c \log n/\epsilon$-stable in view of the assumed SBM. It follows that all graphs that satisfies $\hat{C}$-concentration are at least $c \log n/\epsilon$-stable in view of the assumed SBM. Applying Lemma 1, $\mathcal{M}^f_{\text{Stbl ORACLE}}$ (Algorithm 3) achieves exact recovery in the same conditions. □

Lemma 29. *If $\mathcal{M}^f_{\text{Stbl}}$ (Algorithm 1) achieves exact recovery (under some specific conditions of the SBMs under the view of Lemma 1), Algorithm 3 takes $O(poly(n))$ times w.h.p..*

Proof. Using the same arguments as in Lemma 28, the input graph G satisfies $c \log n/\epsilon$-stable in view of the assumed SBM. It means that w.h.p., $\hat{d}(G)$ is set to $c \log n/\epsilon$ in line 6 instead of going through the computation of $d(G)$ in line 9. Since checking the concentration conditions takes $O(poly(n))$ times, and solving the SDP Relaxation at line 1 takes $O(poly(n))$ times, it follows that w.h.p., Algorithm 3 takes $O(poly(n))$. □

Lemma 30. *(Proof in Appendix B of (Hajek et al., 2016b)) Given G generated by a BASBM with $\rho \leq 0.5$, Algorithm 4 outputs $(\widehat{a}, \widehat{b}) = (a, b) + o(1)$ w.h.p..*

Lemma 31. *If $\mathcal{M}^f_{\text{Stbl}}$ (Algorithm 1) achieves exact recovery (under some specific conditions of the SBMs under the view of Lemma 1), Algorithm 2 ($\mathcal{M}^f_{\text{Stbl FAST}}$) is the same with Algorithm 3 ($\mathcal{M}^f_{\text{Stbl ORACLE}}$) w.h.p..*

Proof. Algorithm 2 is identical with Algorithm 3, except in two steps that Algorithm 3 invokes the oracles.

For the first oracle in line 2, we replace ORACLE_{σ^*} by $\hat{Y}$, which is our estimation of σ^* by the solving the SDP Relaxation. It is clear that under our assumption of exact recovery, $\hat{Y} = \sigma^*(\sigma^*)^T$ w.h.p.. Therefore using $\hat{Y}$ is as good as asking ORACLEw.h.p.

For the second oracle in line 3, we replace $\text{ORACLE}^{\alpha}_{a,b}$ by $\widehat{a}, \widehat{b}$ estimated by Algorithm 4, derived from (Hajek et al., 2016b). Due to Lemma 30, $(\widehat{a}, \widehat{b}) = (a, b) + o(1)$ w.h.p.. It means that when n is large enough, $\widehat{a}, \widehat{b}$ is at least as good as $\text{ORACLE}^{\alpha}_{a,b}$ and the Lemma follows.

□

Algorithm 4 $ParamEstimate(G)$: Algorithm to estimate a, b for BASBM

Let A be the adjacency matrix of the input graph G
$d_i \leftarrow \sum_j A_{ij}$
$w_i \leftarrow \frac{d_i}{\log n}$
$\widehat{w} \leftarrow \frac{1}{n}\sum_i w_i$
$\hat{\rho} \leftarrow \frac{1}{n}\sum \mathbf{1}_{\{w_i \le \widehat{w}\}}$
$\widehat{w}_+ \leftarrow \frac{1}{n}\sum w_i \mathbf{1}_{\{w_i \ge \widehat{w}\}}$
$\widehat{w}_- \leftarrow \frac{1}{n}\sum w_i \mathbf{1}_{\{w_i \le \widehat{w}\}}$
$\widehat{a} \leftarrow \frac{(1-\widehat{\rho})\widehat{w}_+ - \widehat{\rho}\widehat{w}_-}{1-2\widehat{\rho}}$
$\widehat{b} \leftarrow \frac{(1-\widehat{\rho})\widehat{w}_- - \widehat{\rho}\widehat{w}_+}{1-2\widehat{\rho}}$
Return $\widehat{a}, \widehat{b}$

E. General Structure SBM (GSSBM)

In the General Structure SBM (GSSBM), there are multiple possibly unequal communities (i.e., $r > 0$ clusters) and some outliers. The cluster $k^{th}, k \in [r]$, or C_k has size $K_k = \rho_k \times n$ as $n \to \infty$. For any $i > j$, assume that $\rho_i \ge \rho_j > 0$. The are $n - \sum_{k\in[r]} K_k$ outlier vertices do not belong to any cluster. We use $k = 0$ (e.g., in C_0, K_0) to refer to the outliers. Similar to other SBM, we consider the dense regime, where edges are generated with probability $\Omega(\log n/n)$. The ground truth community matrix $Z^* = \sum_{k\in[r]} \xi_k^*(\xi_k)^T$ where ξ_k is the indicator vector of community C_k.

Definition 11. *A graph G with r communities indicated by vectors $\xi_i, i \in [r]$. Edges whose endpoints from a same cluster are generated with probability $p = \frac{a\log n}{n}$ and other edges are generated with probability $q = \frac{b\log n}{n}$ for some constants $a > b > 0$.*

In the non-private setting, the community matrix Z can be obtained by solving the following SDP Relaxation. We use $SDP(G)$ to denote the function taking input G and outputting the optimal solution of the SDP Relaxation.

Definition 12. *The SDP Relaxation of GSSBM.*

$$\hat{Z}_{SDP} = arg\ max \langle A, Z\rangle \tag{229}$$
$$s.t.\ Z \succcurlyeq 0 \tag{230}$$
$$Z_{ii} \le 1\ for\ i \in [n] \tag{231}$$
$$Z_{ij} \ge 0\ for\ i, j \in [n] \tag{232}$$
$$\langle I, Z\rangle = \sum_{k=1}^{r} K_k \tag{233}$$
$$\langle J, Z\rangle = \sum_{k=1}^{r} K_k^2 \tag{234}$$

Definition 13. *We define some quantities used in our analyses:*

- $A(G)$ *is the adjacency matrix of graph G. We drop the parameter when the context is clear.*
- $e(i, C_k)$ *is the number of edges between a node i and nodes from cluster C_k*
- $k(i)$ *is i's cluster*
- $s_i = e(i, C_{k(i)})$
- $\tilde{\tau} = b + 2c_2$
- $\lambda^* = \tilde{\tau}\log n/n$
- $d_i^* = \begin{cases} s_i - \|A - \mathbb{E}[A]\|_2 - \lambda^* K_k\ for\ i \in C_k, k \in [r] \\ 0\ for\ i \in C_0 \end{cases}$

- $D^* = diag\{d^*\}$
- $B^* \in \mathcal{S}^n, B^*_{ij} = \begin{cases} \lambda^* + \frac{e(C_{k(i)},C_{k(j)})}{K_{(k(i))}K_{(k(j))}} - \frac{e(i,C_{k(j)})}{K_{k(j)}} - \frac{e(j,C_{k(i)})}{K_{k(i)}} \textit{ for } k(i) \neq k(j), k(i), k(j) \in [r] \\ \lambda^* - \frac{e(i,C_{k(j)})}{K_{k(j)}} \textit{ for } k(i) = 0, k(j) \in [r] \\ \lambda^* - \frac{e(j,C_{k(i)})}{K_{k(i)}} \textit{ for } k(j) = 0, k(i) \in [r] \\ 0 \textit{ for } k(i) = k(j) \end{cases}$
- $E_r = span\{\xi_1, \xi_2, \cdots, \xi_r\}$
- $\mathbb{E}[A]$ *is a constant matrix that denotes the expected values of* A *over the randomness of the SBM process.* $\mathbb{E}[A] = (p-q)Z^* - pI^{(in)} - qI^{(out)} + qx^T Jx$
- $\rho_{k,k\in[r]} = K_k/n$
- $\rho_{min} = min_{k\in[r]}\rho_k$
- $I(x,y) = x - y\log\frac{ex}{y}$

Assumptions of parameters

$$I(a, b+2c_2) > 1/\rho_{min} \text{ or } I(a, b+2c_2) = 1/\rho_{min} + \Omega(1) \tag{235}$$
$$I(b, b+c_2-c_3/\rho_{min}) > 1/\rho_{min} \text{ or } I(b, b+c_2-c_3/\rho_{min}) = 1/\rho_{min} + \Omega(1) \tag{236}$$
$$I(b, b+2c_2-c_5/\rho_{min}) > 1/\rho_{min} \text{ or } I(b, b+2c_2-c_5/\rho_{min}) = 1/\rho_{min} + \Omega(1) \tag{237}$$

Definition 14. *Definition of* $\mathcal{C}$*-concentration for GSSBM. Given a graph* G *generated by a* $GSSBM$ *defined is Definition 11.* G *is called* $\mathcal{C}$*-concentrated if there exists a tuple of constants* $(c_1, c_2, c_3, c_4, c_5)$ *such that* G *satisfies these conditions:*

- $\|A(G) - \mathbb{E}[A(G)]\| \leq c_1\sqrt{\log n}$
- $\min_{i\in[n]} s_i \geq (b+2c_2)\rho_{k(i)}\log n$
- $\max_{i\in[n],k:k\neq k(i)} e(i, C_k) \leq (b+c_2)K_k\log n/n - c_3\log n$
- $\min_{i,j:k(i)k(j)[k(i)-k(j)]\neq 0} e(C_{k(i)}, C_{k(j)}) \geq K_{k(i)}K_{k(j)}q - 2\sqrt{K_{k(i)}K_{k(j)}}\sqrt{\log n} - c_4\log n$
- $\max_{i\in C_0} e(i, C_{k:k\neq 0}) \geq \tilde{\tau}K_r\frac{\log n}{n} - c_5\log n$

Lemma 32. *Given a graph* G *generated by a* $GSSBM$ *defined is Definition 11. There exists some constant* c_2 *such that:*

$$\min_{i\in[n]} s_i \geq (b+2c_2)\rho_{k(i)}\log n \tag{238}$$

with probability at least $1 - n^{-\Omega(1)}$.

Proof. Since s_i is the number of edges between i and vertices from the same cluster with i, we have $s_i \sim Binom(K_{k(i)}, \frac{a\log n}{n})$, where $K_{k(i)} = \rho_{k(i)}n$. Applying Lemma 20, with $k_n = (b+2c_2)\log n$, we have:

$$\Pr[s_i \leq (b+2c_2)\rho_{k(i)}\log n] = n^{-\rho_{k(i)}(a-(b+2c_2)\log\frac{ea}{b+2c_2}+o(1))} \tag{239}$$
$$\leq n^{-\rho_{k(i)}I(a,b+2c_2)} \tag{240}$$
$$\leq n^{-1-\Omega(1)}, \tag{241}$$

where the first inequality is because of the definition of I and the last inequality is because of the assumption of parameters. Taking union bound on all vertices i and the Lemma follows.

□

Lemma 33. *Given a graph G generated by a $GSSBM$ defined is Definition 11. There exists some constant c_2, c_3 such that:*

$$\max_{i\in[n],k:k\neq k(i)} e(i, C_k) \leq (b + c_2)K_k - c_3 \log n, \tag{242}$$

with probability at least $1 - n^{-\Omega(1)}$.

Proof. By the definition of $e(i, C_{k:k\neq k(i)})$, it is the number of edges of a vertex i and vertices from a different cluster $k \neq k(i)$, or $e(i, C_{k:k\neq k(i)}) \sim Binom(K_k, \frac{b\log n}{n})$. Applying Lemma 20, with $k_n = (b + c_2 - c_3/\rho_k)\log n$, we have:

$$\begin{aligned}
\Pr[e(i, C_k) \geq k_n] &= \Pr[e(i, C_k) \geq (b + c_2 - c_3/\rho_k)\log n] && (243)\\
&\leq n^{-\rho_k(b-(b+c_2-c_3/\rho_k)\log\frac{eb}{b+c_2-c_3/\rho_k}+o(1))} && (244)\\
&\overset{(a)}{\leq} n^{-\rho_k I(b,b+c_2-c_3/\rho_k)} && (245)\\
&\overset{(b)}{\leq} n^{-1-\Omega(1)}, && (246)
\end{aligned}$$

where:

- (a) is because of the definition of I,
- (b) is because of the assumption of parameters.

Taking union bound on all vertices i and the Lemma follows. □

Lemma 34. *Given a graph G generated by a $GSSBM$ defined is Definition 11. There exists some constant c_4 such that:*

$$\min_{i,j:k(i)k(j)[k(i)-k(j)]\neq 0} e(C_{k(i)}, C_{k(j)}) \geq K_{k(i)}K_{k(j)}q - 2\sqrt{K_{k(i)}K_{k(j)}}\sqrt{\log n} - c_4\log n, \tag{247}$$

with probability at least $1 - n^{-\Omega(1)}$.

Proof. Let $k = k(i), k' = k(j)$. The condition implies that k and k' are two cluster and not outliers.

By the definition, $e(C_k, C_{k'}) = Binom(K_kK_{k'}, q)$. By applying standard Chernoff bound, we have:

$$\Pr[e(C_k, C'_k) \leq (1-\alpha)K_kK_{k'} \times q] \leq e^{-\alpha^2 K_kK_{k'}q/2}. \tag{248}$$

Setting,

$$\alpha = \frac{2\sqrt{K_kK_{k'}}\sqrt{\log n} + c_4\log n}{qK_kK_{k'}}, \tag{249}$$

because $|c_4\log n| << \sqrt{K_kK_{k'}}\sqrt{\log n}$,

$$\begin{aligned}
\alpha &> \frac{\sqrt{K_kK_{k'}}\sqrt{\log n}}{qK_kK_{k'}} && (250)\\
&= \frac{\log n}{q\sqrt{K_kK_{k'}}}. && (251)
\end{aligned}$$

Substituting α to Equation 248, we have:

$$\Pr[e(C_k, C_{k'}) \leq qK_kK_{k'} - 2\sqrt{\log n}\sqrt{K_kK_{k'}} - c_4 \log n] = \Pr[e(C_k, C_{k'}) \leq (1-\alpha)K_kK_{k'} \times q] \tag{252}$$

$$\leq e^{-(\frac{\sqrt{\log n}}{q\sqrt{K_kK_{k'}}})^2 K_kK_{k'}q/2} \tag{253}$$

$$= e^{-\frac{\log n}{2q}} \tag{254}$$

$$= n^{-\frac{1}{2q}} \tag{255}$$

$$< n^{-2-\Omega(1)}, \text{ where } n \text{ is large enough,} \tag{256}$$

Taking the union bound on all k, k', the Lemma follows. □

Lemma 35. *Given a graph G generated by a $GSSBM$ defined is Definition 11. There exists some constant c_5 such that:*

$$\max_{i \in C_0} e(i, C_{k(j)}) \geq \tilde{\tau} K_r \frac{\log n}{n} - c_5 \log n, \tag{257}$$

with probability at least $1 - n^{-\Omega(1)}$.

Proof. By the definition of $e(i, C_{k,k\neq 0})$, it is the number of edges of an outlier i and vertices from a cluster $k \neq 0$, or $e(i, C_k) \sim Binom(K_k, \frac{b \log n}{n})$. Applying Lemma 20, with $k_n = (\tilde{\tau} - c_3/\rho_k) \log n$, we have:

$$\Pr[e(i, C_k) \geq k_n] = \Pr[e(i, C_k) \geq (\tilde{\tau} - c_5/\rho_k) \log n] \tag{258}$$

$$\leq n^{-\rho_k(b - (\tilde{\tau} - c_5/\rho_k)\log \frac{eb}{\tilde{\tau} - c_5/\rho_k} + o(1))} \tag{259}$$

$$\overset{(a)}{\leq} n^{-\rho_k I(b, b+2c_2 - c_5/\rho_k)} \tag{260}$$

$$\overset{(b)}{\leq} n^{-1-\Omega(1)}, \tag{261}$$

where:

- (a) is because of the definition of I,
- (b) is because of the assumption of parameters.

Taking union bound on all vertices i and the Lemma follows. □

Lemma 36. *Given a graph G generated by a $GSSBM$ defined is Definition 11. There exists some tuples of constants $\mathcal{C} = (c_1, c_2, c_3, c_4, c_5)$ such that, G is $\mathcal{C}$-concentrated, with probability at least $1 - n^{-\Omega(1)}$.*

Proof. The **First condition** can be proved using the same arguments and settings as in Lemma 14, in which the edges of G are generated with probabilities $\Omega(\log n/n)$. It means that the exists some constant c_1 such that $\|A - \mathbb{E}[A]\|_2 \leq c_1\sqrt{\log n}$ with probability at least $1 - n^{-\Omega(1)}$. Using union bound on it and the **Second condition** (Lemma 32), **Third condition** (Lemma 33), **Fourth condition** (Lemma 34), and the **Fifth condition** (Lemma 35), there exists some tuples of constants $\mathcal{C} = (c_1, c_2, c_3, c_4, c_5)$ such that G satisfies all the conditions with probability at least $1 - n^{-\Omega(1)}$. It means that with probability $1 - n^{-\Omega(1)}$, G is $\mathcal{C}$-concentrated. □

Lemma 37. *Given a graph G generated by a $GSSBM$ defined is Definition 11. If G is $\mathcal{C}$-concentrated then for every graph $G' : d(G, G') < \frac{c}{\epsilon} \log n$, i.e., G' can be constructed by flipping at most $\frac{c}{\epsilon}$ connections of G, G' is $\mathcal{C}' = (c_1', c_2', c_3', c_4', c_5')$-concentrated, where*

$$c_1' = c_1 + \sqrt{2c/\epsilon} \tag{262}$$
$$c_2' = c_2 - \frac{c}{\epsilon\rho_{min}} \tag{263}$$
$$c_3' = c_3 - \frac{c}{\epsilon} \tag{264}$$
$$c_4' = c_4 + c/\epsilon \tag{265}$$
$$c_5' = c_5 - \frac{c}{\epsilon}. \tag{266}$$

Proof. The **First condition** follows from the same arguments in Lemma 15's First condition. We will prove the remaining conditions.

We denote $s'(), e'()$ as the same $s(), e()$ but with the graph G' instead of G. We note that G and G' differ by at most $c\log n/\epsilon$ edges.

The **Second condition**, we have:

$$s_i' \geq s_i - \frac{c\log n}{\epsilon} \tag{267}$$
$$\geq (b+2c_2)\rho_{k(i)}\log n - \frac{c\log n}{\epsilon} \tag{268}$$
$$= (b+2c_2 - \frac{c}{\epsilon\rho_{k(i)}})\rho_{k(i)}\log n \tag{269}$$
$$\geq (b+2c_2 - \frac{c}{\epsilon\rho_{min}})\rho_{k(i)}\log n \tag{270}$$
$$= (b+2c_2')\rho_{k(i)}\log n, \tag{271}$$

which implies $\min_{i\in[n]} s_i' \geq (b+2c_2')\rho_{k(i)}\log n$.

The **Third condition**, we have:

$$e'(i,C_k) \leq e(i,C_k) + \frac{c\log n}{\epsilon} \tag{272}$$
$$\leq (b+c_2)K_k\log n/n - c_3\log n + \frac{c\log n}{\epsilon} \tag{273}$$
$$\leq (b+c_2)K_k\log n/n - (c_3 - \frac{c}{\epsilon})\log n \tag{274}$$
$$= (b+c_2)K_k\log n/n - (c_3')\log n, \tag{275}$$

which implies $\max_{i\in[n],k:k\neq k(i)} e'(i,C_k) \leq (b+c_2)K_k\log n/n - c_3'\log n$.

The **Fourth condition**, we have:

$$e'(C_{k(i)},C_{k(j)}) \geq e(C_{k(i)},C_{k(j)}) - \frac{c\log n}{\epsilon} \tag{276}$$
$$\geq K_{k(i)}K_{k(j)}q - 2\sqrt{K_{k(i)}K_{k(j)}}\sqrt{\log n} - c_4\log n - \frac{c\log n}{\epsilon} \tag{277}$$
$$\geq K_{k(i)}K_{k(j)}q - 2\sqrt{K_{k(i)}K_{k(j)}}\sqrt{\log n} - (c_4 + \frac{c}{\epsilon})\log n \tag{278}$$
$$\geq K_{k(i)}K_{k(j)}q - 2\sqrt{K_{k(i)}K_{k(j)}}\sqrt{\log n} - c_4'\log n, \tag{279}$$

which implies $\min_{i,j:k(i)k(j)[k(i)-k(j)]\neq 0} e'(C_{k(i)}, C_{k(j)}) \geq K_{k(i)}K_{k(j)}q - 2\sqrt{K_{k(i)}K_{k(j)}}\sqrt{\log n} - c_4 \log n$.

The **Fifth condition**, we have:

$$
\begin{aligned}
e'(i, C_k) &\leq e(i, C_k) + \frac{c \log n}{\epsilon} && (280)\\
&\leq (b + 2c_2)K_k \log n/n - c_5 \log n + \frac{c \log n}{\epsilon} && (281)\\
&\leq (b + 2c_2)K_k \log n/n - (c_5 - \frac{c}{\epsilon}) \log n && (282)\\
&= (b + 2c_2)K_k \log n/n - (c_5') \log n && (283)\\
&= \tilde{\tau} K_k \log n/n - (c_5') \log n, && (284)
\end{aligned}
$$

which implies $\max_{i\in C_0} e'(i, C_{k:k\neq 0}) \geq \tilde{\tau} K_r \frac{\log n}{n} - c_5' \log n$, and complete the proof of the Lemma. □

Lemma 38. *Given graph G generated by the GSSBM in Definition 11 that is $\mathcal{C}$-concentrated, $d_i^* > 0$ for every node i belongs to a community, i.e., $k(i) \neq 0$.*

Proof. Recall that by our choice, d_i^* is define as:

$$
d_i^* = \begin{cases} s_i - \|A - \mathbb{E}[A]\|_2 - \lambda^* K_k \text{ for } i \in C_k, k \in [r] \\ 0 \text{ for } i \in C_0 \end{cases} \tag{285}
$$

where $\lambda^* = (b + 2c_2)\frac{\log n}{n}$. Because G is $\mathcal{C}$-concentrated, $A - \mathbb{E}[A] \leq c_1\sqrt{\log n}$ because of the **First condition** and $\min_{i\in[n]} s_i \geq \lambda^* K_{k(i)} + c_2 \log n$ because of the **Second condition**. We have for node $i : k(i) \neq 0$:

$$
\begin{aligned}
d_i^* &= s_i - \|A - \mathbb{E}[A]\|_2 - \lambda^* K_{k(i)} && (286)\\
&\geq \lambda^* K_{k(i)} + c_2 \log n - c_1\sqrt{\log n} - \lambda^* K_{k(i)} && (287)\\
&= c_2 \log n - c_1\sqrt{\log n} && (288)\\
&> 0 \text{ with } n \text{ large enough}, && (289)
\end{aligned}
$$

and the Lemma follows. □

Lemma 39. *Given graph G generated by the GSSBM in Definition 11 that is $\mathcal{C}$-concentrated, $B_{ij}^* > 0$ for every vertices i and j belong to different communities, i.e., $k(i) \neq k(j)$.*

Proof. Recall that $B^* \in \mathcal{S}^n, B_{ij}^* = \begin{cases} \lambda^* + \frac{e(C_{k(i)}, C_{k(j)})}{K_{(k(i))}K_{(k(j))}} - \frac{e(i, C_{k(j)})}{K_{k(j)}} - \frac{e(j, C_{k(i)})}{K_{k(i)}} \text{ for } k(i) \neq k(j), k(i), k(j) \in [r] \\ \lambda^* - \frac{e(i, C_{k(j)})}{K_{k(j)}} \text{ for } k(i) = 0, k(j) \in [r] \\ \lambda^* - \frac{e(j, C_{k(i)})}{K_{k(i)}} \text{ for } k(j) = 0, k(i) \in [r] \\ 0 \text{ for } k(i) = k(j) \end{cases}$,

then we only care about the first three cases.

In the first case, since G is $\mathcal{C}$-concentrated, we have $e(i, C_{k(j)}) \leq (b + c_2)K_{k(j)} - c_3 \log n$ due to the **Third condition** and

$e(C_{k(i)}, C_{k(j)}) \geq K_{k(i)}K_{k(j)}q - 2\sqrt{K_{k(i)}K_{k(j)}}\sqrt{\log n} - c_4 \log n$ due to the **Fourth condition**, we have: l m

$$B^*_{ij} = \lambda^* + \frac{e(C_{k(i)}, C_{k(j)})}{K_{(k(i))}K_{(k(j))}} - \frac{e(i, C_{k(j)})}{K_{k(j)}} - \frac{e(j, C_{k(i)})}{K_{k(i)}} \tag{290}$$

$$\geq \frac{\tilde{\tau}\log n}{n} - \frac{(b+c_2)K_{k_{min}} - c_3 \log n}{K_{k(i)}} - \frac{(b+c_2)K_{k_{min}} - c_3 \log n}{K_{k(i)}} \tag{291}$$

$$+ \frac{q \times K_{k(i)}K_{k(j)} - 2\sqrt{K_{k(i)}K_{k(j)}}\sqrt{\log n} - c_4 \log n}{K_{k(i)}K_{k(j)}} \tag{292}$$

$$\geq (b + 2c_2 + c_3/\rho_{min} - b - c_2 + c_3/\rho_{min} - c_2 + b)\frac{\log n}{n} - \frac{2\sqrt{\log n}}{\rho_{min} n} - \frac{c_4 \log n}{\rho_{min} n^2} \tag{293}$$

$$= \frac{2c_3}{\rho_{min}}\frac{\log n}{n} - \frac{2\sqrt{\log n}}{\rho_{min} n} - \frac{c_4 \log n}{\rho^2_{min} n^2} \tag{294}$$

$$> 0 \text{ when } n \text{ is large enough.} \tag{295}$$

The second and third cases are similar. W.L.O.G., let $k(i) = 0, k(j) \neq 0$, we have:

$$B^*_{ij} = \lambda^* - \frac{e(i, C_{k(j)})}{K_{k(j)}} \tag{296}$$

$$\geq \frac{(b+2c_2)\log n}{n} - \frac{\tilde{\tau}K_r \log n}{\rho_{min} n^2} + \frac{c_5 \log n}{\rho_{min} n} \tag{297}$$

$$= \frac{c_5 \log n}{\rho_{min} n} \tag{298}$$

$$> 0, \tag{299}$$

where the first inequality is because of the **Fifth condition** of $\mathcal{C}$-concentration that G satisfies as the assumption of the Lemma, which completes the proof and the Lemma follows. □

Lemma 40. *(Lemma 10 of (Hajek et al., 2016b)) Suppose there exists $D^* = diag\{d^*_i\}$ with $d^*_i > 0$ for non-outlier vertices and $d^*_i = 0$ for outlier vertices, $B^* = \mathcal{S}^n$ with $B^* \geq 0$ and $B^*_{ij} > 0$ whenever i and j are in different clusters, $\eta^* \in \mathbb{R}$ and $\lambda^* \in \mathbb{R}$ such that $S^* \stackrel{def}{=} D^* - B^* - A + \eta^* I + \lambda^* J$ satisfies that $S^* \succcurlyeq 0$, $\lambda_{r+1}(S^*) > 0$ where $\lambda_r(S^*)$ is the r^{th} smallest eigenvalue of S^*, and*

$$S^*\xi^*_k = 0 \text{ for } k \in [r], \tag{300}$$

$$B^*_{ij}Z^*_{ij} = 0 \text{ for } i, j \in [n]. \tag{301}$$

Then Z^* is the unique solution to the SDP in Definition 12

Lemma 41. *If a graph G generated by the GSSBM in Definition 11 is $\mathcal{C}$-concentrated, then $SDP(G) = Z^*$, i.e., the ground truth community matrix Z^* is the optimal solution of the SDP Relaxation in Definition 12.*

Proof. We now prove that we can construct S^* that satisfies the requirement of Lemma 40. By that, we provide a certificate deterministically constructed from the $\mathcal{C}$-concentration property that guarantees that the optimal solution SDP Relaxation is the ground-truth community matrix Z^*. In other words, solving the SDP in Definition 12 when the input graph G is $\mathcal{C}$-concentrated gives us the ground-truth community matrix Z^* with probability 1.

Lemma 38 and Lemma 39 show us a way to construct D^* and B^* that satisfies Lemma 40. The remaining part is to prove that $S^* \succcurlyeq 0$. It is equivalent to prove that $x^T S^* x \geq 0$ for $x \in \mathbb{R}^n$ and $x \perp E_r$, where $E_r = span\{\xi_1, cdots, \xi_r\}$.

It is clear that:

$$x^T B^* x = \sum_{k \neq k'} \sum_{i \in C_k} \sum_{j \in C_{k'}} B^*_{ij} x_i x_j \tag{302}$$

$$= 0. \tag{303}$$

Now we analyze $x^T \mathbb{E}[A] x$. With $\mathbb{E}[A] = (p-q)Z^* - pI^{(in)} - qI^{(out)} - qJ$ where $I^{(in)}$ and $I^{(out)}$ are the identity matrices for in-cluster nodes and outliers, i.e., $I^{(in)}_{ii} = 1$ for every $k(i) \neq 0$ and $I^{(out)}_{ii} = 1$ for every $k(i) = 0$. We have:

$$x^T \mathbb{E}[A] x = (p-q) x^T Z^* x - p x^T I^{(in)} x - q x^T I^{(out)} x + q x^T J x \tag{304}$$

$$= -p \sum_{i:k(i)\neq 0} x_i^2 - q \sum_{i:k(i)=0} x_i^2 + q x^T J x, \tag{305}$$

where the last equality is because $x^T Z^* x = 0$ for every $x \perp E_r$.

By the definition of $S^* = D^* - B^* - A + \eta^* I + \lambda^* J$, choosing $\eta = \|A - \mathbb{E}[A]\|_2$, we have:

$$x^T S^* x = x^T D^* x - x^T B^* x - x^T A x + \eta^* x^T I x + \lambda^* x^T J x \tag{306}$$

$$\overset{(a)}{=} x^T D^* x - x^T A x + x^T (A - \mathbb{E}[A]) x + \lambda^* x^T J x \tag{307}$$

$$\geq x^T D^* x - x^T \mathbb{E}[A] x + (b + 2c_2) \frac{\log n}{n} x^T J x \tag{308}$$

$$\overset{(b)}{\geq} x^T D^* x + p \sum_{i:k(i)\neq 0} x_i^2 + q \sum_{i:k(i)=0} x_i^2 - \frac{b \log n}{n} x^T J x + (b + 2c_2) \frac{\log n}{n} x^T J x \tag{309}$$

$$\geq x^T D^* x + p \sum_{i:k(i)\neq 0} x_i^2 + q \sum_{i:k(i)=0} x_i^2 + 2c_2 \frac{\log n}{n} x^T J x \tag{310}$$

$$\geq x^T D^* x + p \sum_{i:k(i)\neq 0} x_i^2 + q \sum_{i:k(i)=0} x_i^2 + 2c_2 \frac{\log n}{n} (\sum_i x_i)^2 \tag{311}$$

$$\geq x^T D^* x \tag{312}$$

$$\overset{(c)}{\geq} 0, \tag{313}$$

where:

- (a) is because $x^T B^* X = 0$ for $x \perp E_r$ (Equation 303) and $\eta^* = \|A - \mathbb{E}[A]\|_2$,
- (b) is because of Equation 305,
- (c) is because of the result of Lemma 38;

and the Lemma follows. □

Lemma 42. *A graph G generated by a GSSBM as in Definition 11 is $\frac{c}{\epsilon} \log n$-stable under $SDP(G)$ with probability at least $1 - n^{-\Omega(1)}$.*

Proof. By Lemma 36, G is $\mathcal{C}$-concentrated with probability at least $1 - n^{-\Omega(1)}$. By Lemma 37, all graphs G' (which include G itself) with distance up to $\frac{c}{\epsilon} \log n$ from G are also $\mathcal{C}'$-concentrated. By Lemma 41, since all graphs G' are C'-concentrated, $SDP(G')$ always output the optimal and unique solution σ^* with probability 1, or $SDP(G) = SDP(G') = Y^*$ for all G'. It follows that G is $\frac{c}{\epsilon} \log n$-stable under SDP with probability at least $1 - n^{-\Omega(1)}$. □

Theorem 8. *A graph G generated by a GSSBM as in Definition 11 , and $I(...) > 1$, M_{SDP} with $\delta = n^{-\Omega(1)}$ exactly recovers the ground-truth community Y^*, i.e.,* $\Pr[M^{SDP}(G) \neq Y^*] = n^{-\Omega(1)}$

Proof. Lemma 42 states the stability property of G under SDP, i.e., G is $\frac{c}{\epsilon}\log n$-stable under SDP. It also implies that $SDP(G) = Y^*$ under the effect of $\mathcal{C}$-concentration. By applying Lemma 6, $\Pr[M^{SDP}(G) \neq SDP(G)] = n^{-\Omega(1)}$ or $\Pr[M^{SDP}(G) \neq Y^*] = n^{-\Omega(1)}$.

The condition for the Theorem is that $\mathcal{C}'$ is a valid constant combination, or c/ϵ satisfies the condition of c_2, c_4, c_5. It means:

$$I(a, b + \frac{2c}{\epsilon\rho_{min}}) > 1/\rho_{min} \tag{314}$$

$$I(b, b + \frac{c}{\epsilon}(\frac{1}{\rho_{min}} - 1)) > 1/\rho_{min} \tag{315}$$

$$I(b, b + \frac{c}{\epsilon}(\frac{2}{\rho_{min}} - 1)) > 1/\rho_{min} \tag{316}$$

□